Performance Evaluation 127–128 (2018) 154–175

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

SRPT for multiserver systems[☆]

Isaac Grosof[*], Ziv Scully, Mor Harchol-Balter

Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, USA

ARTICLE INFO

Article history:
Available online 19 October 2018

ABSTRACT

The Shortest Remaining Processing Time (SRPT) scheduling policy and its variants have been extensively studied in both theoretical and practical settings. While beautiful results are known for single-server SRPT, much less is known for *multiserver SRPT*. In particular, stochastic analysis of the M/G/k under SRPT is entirely open. Intuition suggests that multiserver SRPT should be optimal or near-optimal for minimizing mean response time. However, the only known analysis of multiserver SRPT is in the worst-case adversarial setting, where SRPT can be far from optimal. In this paper, we give the *first stochastic analysis* bounding mean response time of the M/G/k under SRPT. Using our response time bound, we show that multiserver SRPT has *asymptotically optimal* mean response time in the heavy-traffic limit. The key to our bounds is a strategic combination of stochastic and worst-case techniques. Beyond SRPT, we prove similar response time bounds and optimality results for several other multiserver scheduling policies.

1. Introduction

The Shortest Remaining Processing Time (SRPT) scheduling policy and variants thereof have been deployed in many computer systems, including web servers [1], networks [2], databases [3], operating systems [4] and FPGA layout systems [5]. SRPT has also long been a topic of fascination for queueing theorists due to its optimality properties. In 1966, the mean response time for SRPT was first derived [6], and in 1968 SRPT was shown to minimize mean response time both in a stochastic sense and in a worst-case sense [7]. However, these beautiful optimality results and the analysis of SRPT are only known for *single-server* systems. Almost nothing is known for *multiserver* systems, such as the M/G/k, even for the case of just $k = 2$ servers.

The SRPT policy for the M/G/k is defined as follows: at all times, the k jobs with smallest remaining processing time receive service, preempting jobs in service if necessary.

We assume a central queue, meaning any job can be dispatched or migrated to any server at any time, and a preempt–resume model, meaning preemption incurs no cost or loss of work.

It seems believable that SRPT should minimize mean response time in multiserver systems because it gives priority to the jobs which will finish soonest, which seems like it should minimize the number of jobs in the system. However, it was shown in 1997 that SRPT is not optimal for multiserver systems in the worst case [8,9]. That is, one can come up with an adversarial arrival sequence for which the mean response time under SRPT is larger that the optimal mean response time. In fact, the ratio by which SRPT's mean response time exceeds the optimal mean response time can be arbitrarily large [8,9].

[☆] This research was supported by NSF-XPS-1629444, NSF-CMMI-1538204, NSF-CSR-180341, and a 2018 Faculty Award from Microsoft. Additionally, Ziv Scully was supported by an ARCS Foundation scholarship and the NSF Graduate Research Fellowship Grant No. DGE-1745016.

[*] Corresponding author.
E-mail addresses: igrosof@cs.cmu.edu (I. Grosof), zscully@cs.cmu.edu (Z. Scully), harchol@cs.cmu.edu (M. Harchol-Balter).

https://doi.org/10.1016/j.peva.2018.10.001
0166-5316/© 2018 Elsevier B.V. All rights reserved.

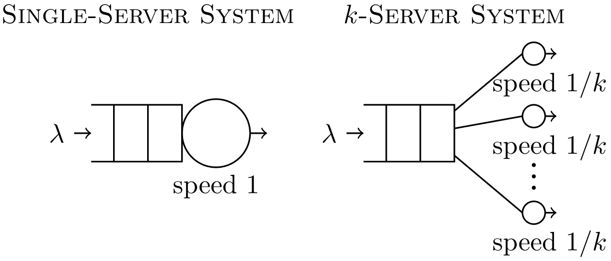


Fig. 1.1. Single-server and k-server systems.

The fact that multiserver SRPT is not optimal in the worst case provokes a natural question about the *stochastic* case.

> *Is SRPT optimal or near-optimal for minimizing mean response time in the* M/G/k*?*

Unfortunately, this question is entirely open. Not only is it not known whether SRPT is optimal, but multiserver SRPT has also eluded stochastic analysis.

> *What is the mean response time for the* M/G/k *under SRPT?*

The purpose of this paper is to answer both of these questions in the high-load setting. Under low load, response time is dominated by service time, which is not affected by the scheduling policy. In contrast, under high load, response time is dominated by queueing time, which can vary wildly under different scheduling policies. We thus focus on the high-load setting, and specifically on the *heavy-traffic limit* as load approaches capacity.

Our main result is that, under mild assumptions on the service requirement distribution,

> *SRPT is an optimal multiserver policy for minimizing mean response time in the* M/G/k*in the heavy-traffic limit.*

We also give the *first mean response time bound for the M/G/k under SRPT*. The bound is valid for all loads and is tight for load near capacity.

In addition to SRPT, we give the *first mean response time bounds for the M/G/k with three other scheduling policies*, specifically Preemptive Shortest Job First (PSJF) [10], Remaining Size Times Original Size (RS) [11,12], and Foreground–Background (FB) [13]. Our bounds imply that in the heavy-traffic limit, under the same mild assumptions as for SRPT above,

- multiserver PSJF and RS are also optimal multiserver scheduling policies; and
- multiserver FB is optimal in the same setting where single-server FB is optimal [14], which is when the service requirement distribution has decreasing hazard rate and the scheduler does not have access to job sizes.

Our approach to analyzing SRPT on k servers is to compare its performance to that of SRPT on a single server which is k times as fast, where both systems have the same arrival rate λ and service requirement distribution S. Specifically, let *SRPT-k* be the policy which uses multiserver SRPT on k servers of speed $1/k$, as shown in Fig. 1.1. Ordinary SRPT on a single server is simply SRPT-1. The *system load* $\rho = \lambda \mathbf{E}[S]$ is the average rate at which work enters the system. The maximal total rate at which the k servers can do work is 1, so the system is stable for $\rho < 1$, which we assume throughout.

Our main result is that in the $\rho \to 1$ limit, the mean response time under SRPT-k, $\mathbf{E}\left[T^{\text{SRPT-}k}\right]$, approaches the mean response time under SRPT-1, $\mathbf{E}\left[T^{\text{SRPT-1}}\right]$. Because SRPT-1 minimizes response time among all scheduling policies, this means that SRPT-k is asymptotically optimal among k-server policies. In particular, let OPT-k be the optimal k-server policy. Then

$$\mathbf{E}\left[T^{\text{SRPT-1}}\right] \le \mathbf{E}\left[T^{\text{OPT-}k}\right] \le \mathbf{E}\left[T^{\text{SRPT-}k}\right],$$

so showing that $\mathbf{E}\left[T^{\text{SRPT-}k}\right]$ approaches $\mathbf{E}\left[T^{\text{SRPT-1}}\right]$ as $\rho \to 1$ also shows that $\mathbf{E}\left[T^{\text{SRPT-}k}\right]$ approaches $\mathbf{E}\left[T^{\text{OPT-}k}\right]$ as $\rho \to 1$.

Specifically, we prove the following sequence of theorems.

Our first theorem is an upper bound on the mean response time of a job of size x under SRPT-k, written $\mathbf{E}\left[T^{\text{SRPT-}k}(x)\right]$. As in the classic SRPT-1 analysis [6], the response time of a job of size x depends on the system load contributed by jobs of size at most x, written $\rho_{\le x}$ (see Definition 3).

Theorem 2. *In an M/G/k, the mean response time of a job of size x under SRPT-k is bounded by*

$$\mathbf{E}\left[T^{\text{SRPT-}k}(x)\right] \le \frac{\int_0^x \lambda t^2 f_S(t)\,dt}{2(1-\rho_{\le x})^2} + \frac{k\rho_{\le x} x}{1-\rho_{\le x}} + \int_0^x \frac{k}{1-\rho_{\le t}}\,dt,$$

where $f_S(\cdot)$ is the probability density function of the service requirement distribution S.

The bound given in Theorem 2 holds for any load ρ and any service requirement distribution S. We use this bound to prove that, under mild conditions on S, the performance of SRPT-k approaches that of SRPT-1 in the $\rho \to 1$ limit, which implies asymptotic optimality of SRPT-k.

Theorem 3. *In an M/G/k with any service requirement distribution S which is either (i) bounded or (ii) unbounded with a tail function which has upper Matuszewska index*[1] *less than* -2,

$$\lim_{\rho \to 1} \frac{\mathbf{E}\left[T^{\text{SRPT-}k}\right]}{\mathbf{E}\left[T^{\text{SRPT-}1}\right]} = 1.$$

The technique by which we bound response time under SRPT-k is widely generalizable. We also use it to give mean response time bounds and optimality results for PSJF-k, RS-k, and FB-k (see Section 7).

Our approach is inspired by two very different worlds: the stochastic world and the adversarial worst-case world. Purely stochastic approaches are difficult to generalize to the M/G/k for many reasons, including the fact that multiserver systems are not work-conserving. Purely adversarial worst-case analysis is easier but leads to weak bounds when directly applied to the stochastic setting. For instance, Leonardi and Raz [8,9] show that for an adversarial arrival sequence, SRPT-k has worse mean response time than the optimal offline k-server policy by a factor of $\Omega(\log(\min(n/k, P)))$, where n is the total number of jobs in the arrival sequence and P is the ratio of the largest job size to the smallest job size. This factor can be arbitrarily large in the context of the M/G/k, because $n \to \infty$ if the arrival sequence is an infinite Poisson process, and $P \to \infty$ if the service requirement distribution is unbounded or allows for arbitrarily small jobs.

What makes our analysis work is a strategic combination of the stochastic and worst-case techniques. We use the more powerful stochastic tools where possible and use worst-case techniques to bound variables for which exact stochastic analysis is intractable.

2. Prior work

Countless papers have been published on the stochastic analysis of the SRPT policy in the single-server model over the last 52 years, beginning in 1966 with Schrage and Miller's response time analysis of the M/G/1 queue under SRPT [6], which was followed shortly by the proof of SRPT's optimality [7]. SRPT remains a major topic of study today. There have been beautiful works on analyzing the tail of response time [15–17], the fairness of SRPT [10,18] and SRPT in different models, such as energy-aware control [19].

However, all of these works analyze *single-server* SRPT. We give the first analysis of multiserver SRPT. While single-server SRPT minimizes mean response time, multiserver SRPT does not[2] [8,9]. We show that multiserver SRPT approaches optimality in heavy traffic.

2.1. Single-server SRPT in heavy traffic

While the exact mean response time analysis of single-server SRPT is known, it is in the form of a triply nested integral. Therefore, it is useful to have a simpler formula for mean response time. Many papers have derived such a formula under heavy traffic [21–24].

Heavy traffic analysis describes the behavior of a queueing system in the limit as load approaches capacity. The most general heavy-traffic analysis of the mean response time of single-server SRPT is due to Lin et al. [21], who characterize the asymptotic behavior of mean response time for general service requirement distributions. They consider three categories of service requirement distributions and give an asymptotic analysis of the mean response time of each:

- bounded distributions,
- distributions whose tail has upper Matuszewska index[3] less than -2, and
- distributions whose tail has lower Matuszewska index greater than -2.

The first and second categories above roughly correspond to the distribution having finite variance, while the third roughly corresponds to the distribution having infinite variance.

In this paper, we restrict our heavy-traffic results to the first two categories, focusing on service requirement distributions that are either bounded or whose tails have upper Matuszewska index less than -2. We build on the work of Lin et al. [21] to give the first heavy-traffic analysis of multiserver SRPT. In particular, we demonstrate that in the heavy-traffic limit, the mean response time of SRPT in a multiserver system with k servers approaches that of SRPT in a single-server system which runs k times faster (see Fig. 1.1).

[1] This technical condition is roughly equivalent to finite variance. See Section 2.1 or Appendix B.

[2] It has been claimed that multiserver SRPT is optimal under the additional assumption that all servers are busy at all times [20, Theorem 2.1]. However, the proof has an error. See Appendix E.

[3] See Appendix B.

2.2. The multiserver priority queue

While there is no existing stochastic analysis of multiserver SRPT, there is some analysis of multiserver priority queues. In a multiserver priority queue, it is assumed that there are finitely many classes of jobs (typically two) with exponential or phase-type service requirement distributions. Thus, the system can be modeled as a multidimensional Markov chain. Mitrani and King [25] give an exact analysis of the two class multiserver system with preemptive priority between the job classes and exponential service times within each class. Sleptchenko et al. [26] extend this analysis to hyperexponential service requirement distributions, and Harchol-Balter et al. [27] extend it further still to support phase-type service requirement distributions and any constant number of preemptive priority classes. However, the solutions found through these extensions can take a very long time to calculate, requiring more time with every added server, priority class, or state in the phase-type distribution.

Our analysis goes beyond the multiclass setting by handling an arbitrary service requirement distribution and a policy, namely SRPT-k, with an infinite set of priorities. Furthermore, our analysis produces a *closed-form* result, in contrast to the numerical results of these prior works.

2.3. Multiserver SRPT in the worst case

While stochastic analysis of multiserver SRPT is open, multiserver SRPT has been well studied in the worst-case setting. Worst-case analysis considers an *adversarially chosen* sequence of job arrival times and service requirements. An online policy (which does not know the arrival sequence) such as SRPT-k is typically compared to the optimal offline policy (which knows the arrival sequence). In the worst-case setting, a policy is a c-approximation if its mean response time is at most c times the mean response time of the offline optimal policy on any arrival sequence.

Leonardi and Raz [8,9] analyze SRPT-k in the worst-case setting under the assumptions that (1) there are n jobs in the arrival sequence and (2) the ratio of the largest and smallest service requirements in the arrival sequence is P. They show that SRPT-k is an $O(\log(\min(n/k, P)))$-approximation for mean response time, where n is the total number of jobs. They also show that any online policy is at least an $\Omega(\log(\min(n/k, P)))$-approximation. This shows that no online policy has a better approximation ratio than SRPT-k by more than a constant factor.

Unfortunately, directly applying the $O(\log(\min(n/k, P)))$ bound on SRPT-k to the M/G/k is not helpful for two reasons. First, the arrival process is an infinite Poisson process, so $n \to \infty$. Second, often the maximum job size is unbounded or the minimum job size is arbitrarily small, so $P \to \infty$ as well.

SRPT has also been considered in other multiserver models. For example, Avrahami and Azar [28] analyze the immediate dispatch setting, in which each server has a queue and jobs are dispatched to these queues on arrival. Each server can only serve the jobs in its queue, and jobs cannot migrate between queues. Within each queue, jobs are served according to SRPT. Avrahami and Azar [28] give a dispatch policy called IMD which achieves the same $O(\log(\min(n/k, P)))$-approximation as SRPT-k, even when compared to the optimal offline policy with migrations. Again, directly applying this to the M/G/k is problematic because $n \to \infty$ and $P \to \infty$.

In contrast with these worst-case results, we show that in the stochastic setting, SRPT-k is asymptotically optimal policy for mean response time in the heavy-traffic limit. Our result holds for an extremely general class of service requirement distributions, including distributions which are unbounded and/or have arbitrarily small jobs.

2.4. Other prior work

Gong and Williamson [29] propose a *single-server* policy called K-SRPT which is superficially similar to our SRPT-k. Specifically, K-SRPT shares the processor between the k jobs in the system with least remaining time. That is, K-SRPT is a hybrid of processor sharing (PS) and SRPT. Crucially, when fewer than k jobs are in the system, K-SRPT allows each job to receive an increased share of the maximum service rate, ensuring work conservation. In contrast, our SRPT-k model never allows a job to receive more than $1/k$ of the maximum service rate of the system, since a job cannot run on more than one server at once. This means SRPT-k is not work-conserving, which makes it difficult to analyze.

3. Model

We study scheduling policies for the M/G/k queue. We write λ for the arrival rate, S for the service requirement distribution, and k for the number of servers. The rate at which any given server completes work is $1/k$. That is, a job with a service requirement, or *size*, of x needs to be served for time kx to complete. The k servers all together have total service rate 1.

The *load* of the M/G/k system, namely the average rate at which work arrives, is

$$\rho = \lambda \mathbf{E}[S].$$

That is, jobs arrive at rate λ jobs per second, each contributing $\mathbf{E}[S]$ work in expectation. We can view $\mathbf{E}[S] = 1/(k\mu)$, where $1/\mu$ is the expected amount of time a job needs to be served to complete. We assume a stable system, meaning $\rho < 1$, and a preempt–resume model, meaning that preemption incurs no cost or loss of work.

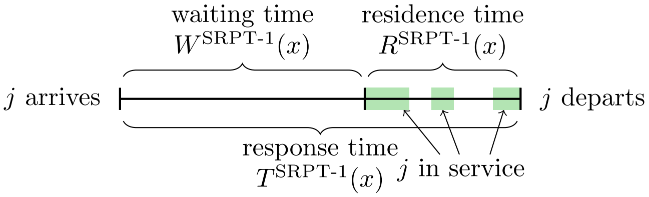


Fig. 4.1. Response time of the tagged job j of size x is the sum of waiting time and residence time.

We will analyze systems in the *heavy-traffic limit*, which is the limit as $\rho \to 1$. More precisely, this is the limit as $\lambda \to 1/\mathbf{E}[S]$ for fixed S.

We analyze and compare systems with $k = 1$ and general k. An example of each is shown in Fig. 1.1. Note that in our model, the M/G/1 and M/G/k systems have the same load ρ.

The primary policy we study is the SRPT-k policy, which is the Shortest Remaining Processing Time policy on k servers. At every moment in time, SRPT-k serves the k jobs with smallest remaining processing time. If there are fewer than k jobs in the system, every job receives service, which leaves some servers idle. Note SRPT-1 is the usual single-server SRPT policy.

4. Background and challenges

Our approach to analyzing response time under SRPT-k is to compare it with SRPT-1. As such, we begin this section by briefly reviewing the analysis of SRPT-1, specifically focusing on the definitions and formulations that will come up in the SRPT-k analysis. We then outline why the SRPT-1 analysis does not easily generalize to SRPT-k with $k > 1$ servers.

4.1. SRPT-1 tagged job tutorial

We now review the technique used by Schrage and Miller [6] to analyze SRPT-1. Consider a particular "tagged" job j, of size x, arriving to a *random system state* drawn from the system's steady-state distribution. We denote j's response time by $T^{\text{SRPT-1}}(x)$. Of course, $T^{\text{SRPT-1}}(x)$ is a random variable which depends on both the random arrivals that occur after j and the random queue state that j observes upon its own arrival.

We split the analysis of $T^{\text{SRPT-1}}(x)$ into two parts, shown in Fig. 4.1:

- *waiting time* $W^{\text{SRPT-1}}(x)$, the time between j's arrival and the moment j first enters service; and
- *residence time* $R^{\text{SRPT-1}}(x)$, the time between the moment j first enters service and j's departure.

Given waiting time and residence time, response time is simply

$$T^{\text{SRPT-1}}(x) = W^{\text{SRPT-1}}(x) + R^{\text{SRPT-1}}(x).$$

Under SRPT-1, j has priority over all jobs with larger remaining size than itself, so such jobs do not impact j's response time.

Definition 1. Suppose job j has remaining size x. A job ℓ is *relevant* to job j if ℓ has remaining size at most x. Otherwise ℓ is *irrelevant* to j.

In particular, we will often consider which jobs are relevant to the tagged job j. We will simply call jobs "relevant" and "irrelevant" when the comparison is clear from context. For the purpose of analyzing j's response time, we can ignore all jobs which are irrelevant to j.

During j's waiting time, the server is only doing *relevant work*, namely work that is due to a relevant job. The total amount of work done is the sum of

- relevant work due to relevant jobs that were in the system when j arrived and
- relevant work due to relevant jobs that arrived after j.

To analyze j's waiting time, we make use of a concept called a "busy period".

Definition 2. A *busy period* started by (possibly random) amount of work V, written $B(V)$, is the amount of time it takes for a single-server work-conserving system that starts with V work to become empty.

Busy periods are very useful because their length depends only on the initial amount of work and the arrival process, not on the service policy or the number of jobs in the system.

In the SRPT-1 system, we do not have to wait for the system to become completely empty for j to start receiving service. We only have to wait for the system to become empty of relevant work. We capture this with the concept of a "relevant busy period".

Definition 3. A *relevant busy period* for a job of size x started by (possibly random) amount of work V, written $B_{\le x}(V)$, is the amount of time it takes for a work-conserving system that starts with V work to become empty, where only arrivals of size at most x, the relevant arrivals, are admitted to the system. A relevant busy period has expectation

$$\mathbf{E}\left[B_{\le x}(V)\right] = \frac{\mathbf{E}[V]}{1-\rho_{\le x}}.$$

Above, $\rho_{\le x}$ is the *relevant load* for a job of size x, which is the total load due to relevant jobs. Its value is

$$\rho_{\le x} = \lambda \mathbf{E}\left[S\mathbb{1}(S \le x)\right],$$

where $\mathbb{1}(\cdot)$ is the indicator function.

This means j's waiting time is a relevant busy period started by the amount of relevant work that the tagged job j sees on arrival. By the PASTA property (Poisson Arrivals See Time Averages) [30], the distribution for the amount of relevant work j sees is the steady-state distribution.

Definition 4. The *steady-state relevant work* for a job of size x under SRPT-1, written $\texttt{RelWork}_{\le x}^{\text{SRPT-1}}$, is the sum of remaining sizes of all jobs with remaining size at most x observed at a random point in time. (An analogous definition applies to SRPT-k.)

By the above discussion, j's waiting time is

$$W^{\text{SRPT-1}}(x) = B_{\le x}\left(\texttt{RelWork}_{\le x}^{\text{SRPT-1}}\right).$$

The analysis of $\texttt{RelWork}_{\le x}^{\text{SRPT-1}}$ is known [6] but outside the scope of this tutorial.

The residence time of j can be analyzed in a similar way. At the start of j's residence time, the SRPT-1 policy serves j, so j, which has remaining x, must be the job with the smallest remaining size in the system. This means the system is effectively empty from j's perspective, because all work relevant to j is gone.

The only work that will be done from this point until j completes is work on j itself and relevant arrivals. Because j's residence time starts with its own work x and ends when that work is done, we can stochastically upper bound j's residence time as a relevant busy period:

$$R^{\text{SRPT-1}}(x) \le_{\text{st}} B_{\le x}(x).$$

The reason this bound is not tight is because j's remaining size decreases during service, which changes the cutoff for relevant jobs. An exact analysis of $R^{\text{SRPT-1}}(x)$ is known [6] but outside the scope of this tutorial.

4.2. Why the tagged job analysis is hard for SRPT-*k*

Having summarized the analysis of SRPT-1, it is natural to ask: why does a similar strategy not work for SRPT-k? The primary difficulty is that multiserver systems are *not work-conserving*, which manifests in two ways.

First, analyzing *busy periods* relies on work conservation, namely the fact that the server is doing work at rate 1 whenever the system is not empty. This allows for many simplifications. For instance, in Definition 3, we define busy periods as being started as a total amount of work, without worrying exactly how that work is divided among jobs. In a k-server system, work is only done at rate 1 if there are k or more jobs in the system. Thus, the exact rate at which work is done varies over time depending on the number of jobs in the system, making it difficult to analyze.

Second, analyzing the *steady-state relevant work* relies on work conservation. The analysis of $\texttt{RelWork}_{\le x}^{\text{SRPT-1}}$ by Schrage and Miller [6] relies on being able to equate $\texttt{RelWork}_{\le x}^{\text{SRPT-1}}$ to the total work in a simpler first-come-first-served system. Equality of remaining work only holds if both systems are work-conserving. The fact that SRPT-k is not work-conserving means that we cannot make such an argument.

5. Analysis of SRPT-*k*

As explained in Section 4.2, traditional tagged job analysis cannot be applied to SRPT-k because SRPT-k is not work-conserving. Our approach is to find a way to make SRPT-k appear work-conserving while the tagged job j is in the system. We do this by introducing the new concept of *virtual work*. Virtual work encapsulates all of the time that the servers spend either idle or working on irrelevant jobs while j is in the system. By thinking of these times as "virtual work", the system appears to be work-conserving while j is in the system, allowing us to bound the response time of j.

Consider a tagged job j of size x. Recall from Definition 1 that only jobs of remaining size at most x are *relevant* to j when j arrives. We will bound j's response time by bounding the *total amount of server activity* between j's arrival and departure. Between j's arrival and departure, each server can be doing one of four categories of work.

- *Tagged work:* serving j.
- *Old work:* serving a job which is relevant to j that was in the system upon j's arrival.
- *New work:* serving a job which is relevant to j that arrived after j.
- *Virtual work:* either idling or serving a job which is irrelevant to j.

The response time of j is exactly the total of tagged, old, new, and virtual work. The main idea behind our analysis is to bound this total by a single (work-conserving) relevant busy period (see Definition 3). The busy period is still defined with regard to a single server system, making the analysis straightforward.

We already know a few facts about the four categories of work.

- Tagged work is j's size x.
- Old work is equal to the amount of relevant work seen by j upon arrival.[4] By the PASTA property [30], this is $\texttt{RelWork}_{\le x}^{\text{SRPT-}k}$, the steady state amount of relevant work for a job of size x (see Definition 4).
- New work is bounded by all jobs which are relevant to a job of remaining size x that arrive during a relevant busy period $B_{\le x}(\cdot)$ started by tagged, old, and virtual work.[5] This is only an upper bound because we ignore the fact that j's remaining size decreases as j is served, which changes the size cutoff for relevant jobs.
- Virtual work is as of yet unknown. We denote with the random variable $\texttt{VirtWork}^{\text{SRPT-}k}(x)$ the amount of virtual work done while j is in the system.

Taken together, these yield the bound

$$T^{\text{SRPT-}k}(x) \le_{\text{st}} B_{\le x}\left(x + \texttt{RelWork}_{\le x}^{\text{SRPT-}k} + \texttt{VirtWork}^{\text{SRPT-}k}(x)\right). \tag{5.1}$$

Our task in the remainder of this section is to bound $\texttt{RelWork}_{\le x}^{\text{SRPT-}k}$ and $\texttt{VirtWork}^{\text{SRPT-}k}(x)$ as tightly as we can. We use worst-case methods to bound $\texttt{VirtWork}^{\text{SRPT-}k}(x)$ and a combination of stochastic and worst-case methods to bound $\texttt{RelWork}_{\le x}^{\text{SRPT-}k}$.

5.1. Virtual work

We start by bounding $\texttt{VirtWork}^{\text{SRPT-}k}(x)$, the virtual work done while j is in the system. A purely stochastic analysis of virtual work would be very difficult. Fortunately, a simple worst-case bound suffices for our purposes. The key is that a server can do virtual work *only while j is in service at a different server*. This is because SRPT-k never allows an irrelevant job to have priority over j.

Lemma 1. *The virtual work is bounded by*

$$\texttt{VirtWork}^{\text{SRPT-}k}(x) \le (k-1)x.$$

Proof. Virtual work only occurs while j is in service. The maximum possible virtual work is achieved by all $k-1$ other servers doing virtual work whenever j is in service. Each server does work at rate $1/k$. This means j is in service for time kx, during which virtual work is done at rate at most $(k-1)/k$. □

5.2. Relevant work

Our next task is to bound $\texttt{RelWork}_{\le x}^{\text{SRPT-}k}$, the steady state amount of relevant work for a job of size x under SRPT-k. As with virtual work, a purely stochastic analysis of relevant work would be very difficult. We therefore take the following hybrid approach. We consider a pair of systems, one using SRPT-1 and the other using SRPT-k, experiencing the same arrival sequence. We compare the amounts of relevant work in each system, giving a *worst-case bound for the difference*. This allows us to use the previously known *stochastic analysis* of $\texttt{RelWork}_{\le x}^{\text{SRPT-}1}$ to give a stochastic bound for $\texttt{RelWork}_{\le x}^{\text{SRPT-}k}$.

Consider running a pair of systems under the same job arrival sequence:

- *System* 1, which schedules using SRPT-1; and
- *System* k, which schedules using SRPT-k.

For any time t, let $\texttt{RelWork}_{\le x}^{(1)}(t)$ be the amount of relevant work in System 1 at t, and similarly for $\texttt{RelWork}_{\le x}^{(k)}(t)$. Our goal is to give a worst-case bound for the difference in relevant work between Systems 1 and k,

$$\Delta_{\le x}(t) = \texttt{RelWork}_{\le x}^{(k)}(t) - \texttt{RelWork}_{\le x}^{(1)}(t).$$

To bound $\Delta_{\le x}(t)$, we split times t into two types of intervals:

- *few-jobs intervals*, during which there are fewer than k relevant jobs at a time in System k; and
- *many-jobs intervals*, during which there are at least k relevant jobs at a time in System k.

A similar type of splitting was used by Leonardi and Raz [8,9].

[4] One might worry that an *old job* that is irrelevant when j arrives could later become relevant to j, and therefore be part of old work, but this does not occur under SRPT-k.

[5] One might worry that a *new job* that is irrelevant when it arrives could later become relevant to j, and therefore be part of new work, but this does not occur under SRPT-k.

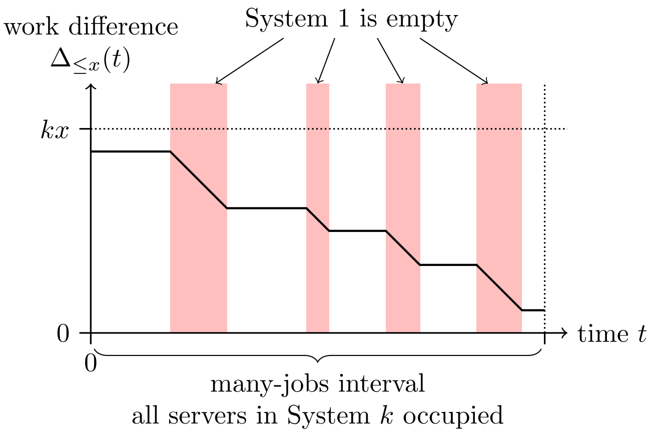


Fig. 5.1. Relevant work difference is nonincreasing during many-jobs intervals.

As a reminder, a job is *relevant* if its remaining size is at most x and *irrelevant* otherwise (see Definition 1). Note that many-jobs intervals are defined only in terms of System k, so System 1 may or may not have relevant jobs during a many-jobs interval.

Lemma 2. *For any arrival sequence and at any time t, the difference between the relevant work in System* 1 *and the relevant work in System k is bounded by*

$$\Delta_{\leq x}(t) \leq kx.$$

Lemma 2 shows that $\Delta_{\leq x}(t)$ is bounded at all times. We can summarize the proof of Lemma 2 as follows. In a few-jobs interval, $\Delta_{\leq x}(t)$ is bounded because there are few relevant jobs in System 1 and each contributes a bounded amount of relevant work. In a many-jobs interval, $\Delta_{\leq x}(t)$ is nonincreasing, and hence bounded.

One might intuitively expect $\Delta_{\leq x}(t)$ to be constant during a many-jobs interval. However, $\Delta_{\leq x}(t)$ can decrease during a many-jobs interval, namely when System 1 is empty, as shown in Fig. 5.1.

Since the few-jobs and many-jobs intervals cover all possible times, Lemma 2 always holds.

Proof of Lemma 2. Any time t is in either a few-jobs interval or a many-jobs interval. The case where t is in a few-jobs interval is simple: there are at most $k-1$ relevant jobs in System k at time t, each of remaining size at most x, so

$$\Delta_{\leq x}(t) \leq \texttt{RelWork}^{(k)}_{\leq x}(t) \leq (k-1)x.$$

Suppose instead that t is in a many-jobs interval. Let time s be the start of the many-jobs interval containing t. We will show

$$\Delta_{\leq x}(t) \leq \Delta_{\leq x}(s) \leq kx.$$

We first show that $\Delta_{\leq x}(t) \leq \Delta_{\leq x}(s)$. Let

$$D^{(1)} = \texttt{RelWork}^{(1)}_{\leq x}(t) - \texttt{RelWork}^{(1)}_{\leq x}(s)$$
$$D^{(k)} = \texttt{RelWork}^{(k)}_{\leq x}(t) - \texttt{RelWork}^{(k)}_{\leq x}(s)$$

be the change in relevant work from s to t in Systems 1 and k, respectively. Because

$$\Delta_{\leq x}(t) - \Delta_{\leq x}(s) = D^{(k)} - D^{(1)},$$

it suffices to show $D^{(k)} \leq D^{(1)}$.

We can write $D^{(1)}$ as a sum of three components,

$$D^{(1)} = \texttt{Arrivals}^{(1)} + \texttt{NewlyRelevant}^{(1)} - \texttt{Served}^{(1)},$$

which are defined as follows.

- $\texttt{Arrivals}^{(1)}$ is the relevant work added during $[s, t]$ due to relevant new arrivals.
- $\texttt{NewlyRelevant}^{(1)}$ is the relevant work added during $[s, t]$ due to the server serving irrelevant jobs until they reach remaining size x, at which point they become relevant. For our purposes, all that matters is that $\texttt{NewlyRelevant}^{(1)} \geq 0$.
- $\texttt{Served}^{(1)}$ is the amount of relevant work done by the server during $[s, t]$. System 1 does relevant work at rate 1 if it has any relevant jobs and rate 0 otherwise, so $\texttt{Served}^{(1)} \leq t - s$.

We define analogous quantities for System k and compare them to their System 1 counterparts.

- $\mathtt{Arrivals}^{(k)} = \mathtt{Arrivals}^{(1)}$ because the two systems experience the same arrivals.
- $\mathtt{NewlyRelevant}^{(k)} = 0$ because $[s, t]$ is within a many-jobs interval, during which System k has at least k relevant jobs. Therefore, there is never an opportunity for an irrelevant job to be served and become relevant. In particular,
$$\mathtt{NewlyRelevant}^{(k)} \le \mathtt{NewlyRelevant}^{(1)}.$$
- $\mathtt{Served}^{(k)} = t - s$ because $[s, t]$ is within a many-jobs interval, during which System k has at least k relevant jobs. Therefore, its servers do relevant work at combined rate 1 during all of $[s, t]$. In particular,
$$\mathtt{Served}^{(k)} \ge \mathtt{Served}^{(1)}.$$

The three comparisons above imply $D^{(k)} \le D^{(1)}$, as desired.

All that remains is to show $\Delta_{\le x}(s) \le kx$. Recall that s is the start of a many-jobs interval. There are two ways to enter a many-jobs interval. In both cases, we show that $\Delta_{\le x}(s) \le kx$.

One way a many-jobs interval can start is when *a relevant job arrives while System k has $k-1$ relevant jobs*. The same arrival occurs in System 1, so $\Delta_{\le x}(s) = \Delta_{\le x}(s^-)$, where s^- is the instant before the arrival. But s^- is the end of a few-jobs interval, during which System k has at most $k-1$ relevant jobs, so

$$\Delta_{\le x}(s) = \Delta_{\le x}(s^-) \le \mathtt{RelWork}^{(k)}_{\le x}(s^-) \le (k-1)x.$$

The other way a many-jobs interval can start is for *irrelevant jobs already in System k to become relevant*. For this to happen, System k must be serving $i \ge 1$ irrelevant jobs at s^-. Because relevant jobs have priority over irrelevant jobs, all relevant jobs must also be in service at s^-. There are i irrelevant jobs in service at s^-, so there are at most $k - i$ relevant jobs at s^-. At time s, at most i irrelevant jobs become relevant, so there are at most k relevant jobs at s. Each relevant job has size at most x, so

$$\Delta_{\le x}(s) \le \mathtt{RelWork}^{(k)}_{\le x}(s) \le kx. \quad \square$$

5.3. Response time bound

Theorem 1. *In an M/G/k, the response time of a job of size x under SRPT-k is bounded by*

$$T^{\text{SRPT-}k}(x) \le_{\text{st}} W^{\text{SRPT-1}}(x) + B_{\le x}(2kx),$$

where $W^{\text{SRPT-1}}(x)$ denotes the waiting time of a job of size x under SRPT-1.

Proof. From (5.1), we know that

$$T^{\text{SRPT-}k}(x) \le_{\text{st}} B_{\le x}\Big(x + \mathtt{RelWork}^{\text{SRPT-}k}_{\le x} + \mathtt{VirtWork}^{\text{SRPT-}k}(x)\Big).$$

By plugging in Lemmas 1 and 2, we find that

$$\begin{aligned} T^{\text{SRPT-}k}(x) &\le_{\text{st}} B_{\le x}\big(\mathtt{RelWork}^{\text{SRPT-1}}_{\le x} + 2kx\big) \\ &= B_{\le x}\big(\mathtt{RelWork}^{\text{SRPT-1}}_{\le x}\big) + B_{\le x}(2kx). \end{aligned}$$

Recall from Section 4.1 that the waiting time in SRPT-1 is

$$W^{\text{SRPT-1}}(x) = B_{\le x}\big(\mathtt{RelWork}^{\text{SRPT-1}}_{\le x}\big),$$

giving the desired bound. $\square$

While Theorem 1 gives a good bound on the response time under SRPT-k, we can tighten the bound further by making use of three ideas.

- As the tagged job j is served, its remaining size decreases. This decreases the size cutoff for new arrivals to be relevant, so not as many arriving jobs contribute to new work. Our current bounds do not account for this effect.
- In Lemma 2, we bound the difference $\Delta_{\le x}(t)$ between relevant work in System 1, which uses SRPT-1, and relevant work in System k, which uses SRPT-k. It turns out that the same proof holds when System 1 uses PSJF-1, the preemptive shortest job first policy, instead of SRPT-1. This improves the bound because waiting time under PSJF-1 is smaller than waiting time under SRPT-1 [11].
- Even after replacing SRPT-1 with PSJF-1, Lemma 2 is not tight. In particular, $\Delta_{\le x}(t)$ is at most x times the number of servers serving relevant jobs at time t, and there are not always k such servers.

These ideas allow us to prove the following bound on mean response time, which is strictly tighter than the one given by Theorem 1.

Theorem 2. *In an M/G/k, the mean response time of a job of size x under SRPT-k is bounded by*

$$\mathbf{E}\left[T^{\text{SRPT-}k}(x)\right] \le \frac{\int_0^x \lambda t^2 f_S(t)\,dt}{2(1-\rho_{\le x})^2} + \frac{k\rho_{\le x}x}{1-\rho_{\le x}} + \int_0^x \frac{k}{1-\rho_{\le t}}\,dt,$$

where $f_S(\cdot)$ is the probability density function of the service requirement distribution S.

Proof. See Appendix A. □

Note that the first term of Theorem 2's upper bound is the mean waiting time of a job of size x under PSJF-1.

6. Optimality of SRPT-*k* in heavy traffic

With the bound derived in Theorem 1, we can prove our main result on the optimality of SRPT-k in the heavy-traffic limit. Theorem 3 will refer to $\mathbf{E}\left[T^{\text{SRPT-}k}\right]$, which is derived from Theorem 1 by taking the expectation over possible sizes x.

Theorem 3. *In an M/G/k with any service requirement distribution S which is either (i) bounded or (ii) unbounded with tail function of upper Matuszewska index*[6] *less than* -2,

$$\lim_{\rho\to 1} \frac{\mathbf{E}\left[T^{\text{SRPT-}k}\right]}{\mathbf{E}\left[T^{\text{SRPT-}1}\right]} = 1.$$

To prove Theorem 3, we start with a result from the literature on the performance of SRPT-1 in the heavy-traffic limit [21].

Lemma 3. *In an M/G/1 with any service requirement distribution S which is either (i) bounded or (ii) unbounded with tail function of upper Matuszewska index less than* -2,

$$\lim_{\rho\to 1} \frac{\log\left(\frac{1}{1-\rho}\right)}{\mathbf{E}\left[T^{\text{SRPT-}1}\right]} = 0.$$

Proof. Follows immediately from results of Lin et al. [21]. See Appendix C. □

The next step in proving Theorem 3, is to use the bound on $T^{\text{SRPT-}k}(x)$ provided by Theorem 1. Let $H(x)$ be the bound on $\mathbf{E}\left[T^{\text{SRPT-}k}(x)\right]$,

$$H(x) = \mathbf{E}\left[W^{\text{SRPT-}1}(x) + B_{\le x}(2kx)\right]. \tag{6.1}$$

By taking the expectation of drawing size x from the service requirement distribution S, Theorem 1 implies $\mathbf{E}\left[T^{\text{SRPT-}k}\right] \le \mathbf{E}\left[H(S)\right]$. The following lemma shows that $\mathbf{E}\left[H(S)\right]$ approaches $\mathbf{E}\left[T^{\text{SRPT-}1}\right]$ in the heavy-traffic limit.

Lemma 4. *In an M/G/k with any service requirement distribution S which is either (i) bounded or (ii) unbounded with tail function of upper Matuszewska index less than* -2,

$$\lim_{\rho\to 1} \frac{\mathbf{E}\left[H(S)\right]}{\mathbf{E}\left[T^{\text{SRPT-}1}\right]} = 1.$$

Proof. We know $\mathbf{E}\left[H(S)\right] \ge \mathbf{E}\left[T^{\text{SRPT-}k}\right]$ by Theorem 1, and we know $\mathbf{E}\left[T^{\text{SRPT-}k}\right] \ge \mathbf{E}\left[T^{\text{SRPT-}1}\right]$ by optimality of SRPT-1, so

$$\frac{\mathbf{E}\left[H(S)\right]}{\mathbf{E}\left[T^{\text{SRPT-}1}\right]} \ge 1.$$

We thus only need to show

$$\lim_{\rho\to 1} \frac{\mathbf{E}\left[H(S)\right]}{\mathbf{E}\left[T^{\text{SRPT-}1}\right]} \le 1.$$

Because $W^{\text{SRPT-}1} \le T^{\text{SRPT-}1}$, by (6.1) it suffices to show

$$\lim_{\rho\to 1} \frac{\mathbf{E}\left[B_{\le S}(2kS)\right]}{\mathbf{E}\left[T^{\text{SRPT-}1}\right]} = 0. \tag{6.2}$$

[6] See Section 2.1 or Appendix B.

Applying standard results for busy periods [31],

$$\mathbf{E}\left[B_{\le S}(2kS)\right] = 2k\mathbf{E}\left[\frac{S}{1-\rho_{\le S}}\right] = 2k\int_0^\infty \frac{xf_S(x)}{1-\rho_{\le x}}\,dx,$$

where $f_S(\cdot)$ is the probability density function of S. To compute the integral, we make a change of variables from x to $\rho_{\le x}$ (see Definition 3), which uses the following facts:

$$\begin{aligned}
\rho_{\le x} &= \lambda\mathbf{E}\left[S\mathbb{1}(S<x)\right] = \int_0^x \lambda t f_S(t)\,dt\\
\frac{d\rho_{\le x}}{dx} &= \lambda x f_S(x)\\
\rho_{\le 0} &= 0\\
\lim_{x\to\infty}\rho_{\le x} &= \rho.
\end{aligned}$$

Given this change of variables, we compute

$$\begin{aligned}
\mathbf{E}\left[\frac{S}{1-\rho_{\le S}}\right] &= \int_0^\infty \frac{xf_S(x)}{1-\rho_{\le x}}\,dx\\
&= \int_0^\rho \frac{1}{\lambda(1-\rho_{\le x})}\,d\rho_{\le x}\\
&= \frac{1}{\lambda}\ln\left(\frac{1}{1-\rho}\right)\\
&= \Theta\left(\log\left(\frac{1}{1-\rho}\right)\right).
\end{aligned}$$

This means $\mathbf{E}\left[B_{\le S}(2kS)\right] = \Theta(\log(1/(1-\rho)))$, so (6.2) follows from Lemma 3. □

Armed with Theorem 1 and Lemma 4, we are now prepared to prove our main result, Theorem 3.

Proof of Theorem 3. Because SRPT-1 minimizes mean response time, it suffices to show that

$$\lim_{\rho\to1}\frac{\mathbf{E}\left[T^{\text{SRPT-}k}\right]}{\mathbf{E}\left[T^{\text{SRPT-}1}\right]} \le 1,$$

which follows immediately from Theorem 1 and Lemma 4. □

Theorem 3 and the optimality of SRPT-1 imply that SRPT-k is optimal in the heavy-traffic limit.

Corollary 1. *In an M/G/k with any service requirement distribution S which is either (i) bounded or (ii) unbounded with tail function of upper Matuszewska index less than* -2,

$$\limsup_{\rho\to1}\frac{\mathbf{E}\left[T^{\text{SRPT-}k}\right]}{\mathbf{E}\left[T^{P}\right]} \le 1$$

for any scheduling policy P.

Recall from (6.1) that Theorem 1 implies $\mathbf{E}\left[T^{\text{SRPT-}k}\right] \le \mathbf{E}\left[H(S)\right]$. Similarly, letting

$$I(x) = \frac{\int_0^x \lambda t^2 f_S(t)\,dt}{2(1-\rho_{\le x})^2} + \frac{k\rho_{\le x}x}{1-\rho_{\le x}} + \int_0^x \frac{k}{1-\rho_{\le t}}\,dt,$$

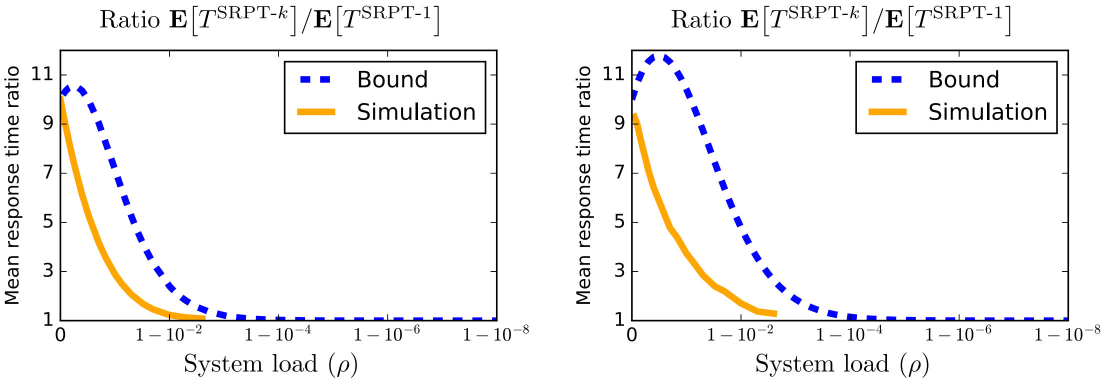
Theorem 2 implies $\mathbf{E}\left[T^{\text{SRPT-}k}\right] \le \mathbf{E}\left[I(S)\right]$. Lemma 4 and the optimality of SRPT-1 imply that these bounds on SRPT-k's mean response time are tight as $\rho \to 1$.

Corollary 2. *In an M/G/k with any service requirement distribution S which is either (i) bounded or (ii) unbounded with tail function of upper Matuszewska index less than* -2,

$$\lim_{\rho\to1}\frac{\mathbf{E}\left[H(S)\right]}{\mathbf{E}\left[T^{\text{SRPT-}k}\right]} = \lim_{\rho\to1}\frac{\mathbf{E}\left[I(S)\right]}{\mathbf{E}\left[T^{\text{SRPT-}k}\right]} = 1.$$

Proof. After applying Theorem 1, Lemma 4, and the optimality of SRPT-1, we know that

$$\lim_{\rho\to1}\frac{\mathbf{E}\left[H(S)\right]}{\mathbf{E}\left[T^{\text{SRPT-}k}\right]} = 1.$$



The plots above show the ratio $\mathbf{E}\left[T^{\text{SRPT-}k}\right]/\mathbf{E}\left[T^{\text{SRPT-1}}\right]$. Observe that as $\rho \to 1$, both our bound and the simulation converge to a ratio of 1. Our simulations of this ratio are the solid orange curves. Our analytic upper bounds derived in Theorem 2 are the dashed blue curves. We use $k = 10$ servers. The service requirement distribution S is Uniform$(0, 2)$ in the left plot and a Hyperexponential distribution with $E[S] = 1$ and $C^2 = 10$ in the right plot. We only simulate up to $\rho = 0.9975$ due to long convergence times.

Fig. 6.1. Convergence of mean response time ratio.

All that remains is to show $I(x) \le H(x)$. This holds because

$$\frac{\int_0^x \lambda t^2 f_S(t)\,dt}{2(1-\rho_{\le x})^2} \le \mathbf{E}\left[W^{\text{SRPT-1}}(x)\right]$$

by the standard analysis of $W^{\text{SRPT-1}}(x)$ [6], and

$$\frac{k\rho_{\le x}x}{1-\rho_{\le x}} + \int_0^x \frac{k}{1-\rho_{\le t}}\,dt \le \frac{2kx}{1-\rho_{\le x}} = \mathbf{E}\left[B_{\le x}(2kx)\right]. \quad \square$$

As an illustration of the optimality of SRPT-k, we plot the ratio $\mathbf{E}\left[T^{\text{SRPT-}k}\right]/\mathbf{E}\left[T^{\text{SRPT-1}}\right]$ in Fig. 6.1. The solid orange lines show simulation results for this ratio. For the dashed blue lines, we used our analysis from Theorem 2 as an upper bound on $\mathbf{E}\left[T^{\text{SRPT-}k}\right]$, and divided by the known results for $\mathbf{E}\left[T^{\text{SRPT-1}}\right]$. The important feature to notice in Fig. 6.1 is that as system load ρ approaches 1, both our analytic bound and the simulation converge to 1.

7. Other scheduling policies

We generalize our analysis to give the first response time bounds on several additional multiserver scheduling policies. Using the bounds, we prove optimality results for each policy as $\rho \to 1$. For each policy P, we generalize the usual M/G/1 policy, written P-1, to a multiserver policy for the M/G/k, written P-k, by preemptively serving the k jobs with highest priority at any time.

- *Preemptive Shortest Job First* (PSJF) prioritizes the jobs with smallest *original* size. PSJF-1 achieves performance comparable to SRPT despite not tracking every job's age [31].
- *Remaining Size Times Original Size* (RS) prioritizes the jobs with the smallest product of original size and remaining size. RS is also known as Size Processing Time Product (SPTP). RS-1 is optimal for minimizing *mean slowdown* [12].
- *Foreground–Background* (FB) prioritizes the jobs with smallest *age*, meaning the jobs that have been served the least so far. FB is also known as Least Attained Service (LAS). When the service requirement distribution has decreasing hazard rate, FB-1 minimizes mean response time among all scheduling policies that do not have access to job sizes [14].

We give the first response time bounds for PSJF-k, RS-k and FB-k. We then use these bounds to prove the following optimality results, under mild assumptions on the service requirement distribution:

- In the $\rho \to 1$ limit, PSJF-k and RS-k minimize mean response time among all scheduling policies (see Theorems 5 and 7).
- In the $\rho \to 1$ limit, FB-k minimizes mean response time under the same conditions as FB-1 (see Theorem 9).

Our analyses follow the same steps as in Section 5.

- Use the four categories of work to bound the response time of the tagged job j in terms of virtual work and steady-state relevant work.

- Bound virtual work.
- Bound steady-state relevant work.

Because different scheduling policies prioritize jobs differently, we use a different definition of "relevant jobs" for each policy. Under PSJF-k and RS-k, the definition of relevant jobs is very similar to that for SRPT-k, allowing us to use familiar tools such as relevant busy periods $B_{\le x}(\cdot)$. However, FB-k uses a somewhat different definition of relevant jobs, resulting in a few changes to the analysis.

Finally, in Section 7.4, we discuss why our technique *does not* generalize to the First-Come, First-Served (FCFS) scheduling policy.

7.1. Preemptive Shortest Job First (PSJF-k)

As usual, we consider a tagged job j of size x. Under PSJF-k, another job ℓ is *relevant* to j if ℓ has *original* size at most x. With this definition of relevance, we divide work into the same four categories as in Section 5, namely tagged, old, new, and virtual. This bounds the response time of j by

$$T^{\text{PSJF-}k} \le_{\text{st}} B_{\le x}\Big(x + \texttt{RelWork}_{\le x}^{\text{PSJF-}k} + \texttt{VirtWork}^{\text{PSJF-}k}(x)\Big). \tag{7.1}$$

The proof of Lemma 1 works nearly verbatim for PSJF-k, so

$$\texttt{VirtWork}^{\text{PSJF-}k}(x) \le (k-1)x. \tag{7.2}$$

The analysis of steady-state relevant work is similar to that in Section 5.2. We consider a pair of systems experiencing the same arrival sequence: System 1, which uses PSJF-1, and System k, which uses PSJF-k. We define $\Delta_{\le x}^{\text{PSJF-}k}(t)$ to be the difference between the amounts of relevant work in the two systems at time t. We then bound $\Delta_{\le x}^{\text{PSJF-}k}(t)$.

Lemma 5. *The difference in relevant work between Systems* 1 *and* k *is bounded by*

$$\Delta_{\le x}^{\text{PSJF-}k}(t) \le (k-1)x.$$

Proof. We define few-jobs intervals and many-jobs intervals as in Section 5.2. The case where t is in a few-jobs interval is simple: there are at most $k-1$ relevant jobs in System k at time t, each of remaining size at most x, so

$$\Delta_{\le x}^{\text{PSJF-}k}(t) \le (k-1)x.$$

Suppose instead that t is in a many-jobs interval. Let time s be the start of the many-jobs interval containing t. By essentially the same argument as in the proof of Lemma 2,[7]

$$\Delta_{\le x}^{\text{PSJF-}k}(t) \le \Delta_{\le x}^{\text{PSJF-}k}(s).$$

It thus suffices to show $\Delta_{\le x}^{\text{PSJF-}k}(s) \le (k-1)x$. The only way a many-jobs interval can start under PSJF-k is for a relevant job to arrive while System k has $k-1$ relevant jobs. The same arrival occurs in System 1, so

$$\Delta_{\le x}^{\text{PSJF-}k}(s) = \Delta_{\le x}^{\text{PSJF-}k}(s^-) \le (k-1)x$$

because s^-, the instant before s, is in a few-jobs interval. □

Theorem 4. *In an M/G/k, the response time of a job of size x under PSJF-k is bounded by*

$$T^{\text{PSJF-}k}(x) \le_{\text{st}} W^{\text{PSJF-}1}(x) + B_{\le x}((2k-1)x).$$

Proof. By (7.1), (7.2), and Lemma 5,

$$\begin{aligned} T^{\text{PSJF-}k}(x) &\le_{\text{st}} B_{\le x}\big(\texttt{RelWork}_{\le x}^{\text{PSJF-}1} + (2k-1)x\big) \\ &= B_{\le x}\big(\texttt{RelWork}_{\le x}^{\text{PSJF-}1}\big) + B_{\le x}(2k-1). \end{aligned}$$

The waiting time in PSJF-1 is

$$W^{\text{PSJF-}1}(x) = B_{\le x}\big(\texttt{RelWork}_{\le x}^{\text{PSJF-}1}\big),$$

giving the desired bound. □

With the bound derived in Theorem 4, we can prove that PSJF-k also minimizes mean response time in the heavy-traffic limit.

[7] In fact, the argument for PSJF is slightly simpler than that for SRPT, because irrelevant jobs never become relevant under PSJF.

Theorem 5. *In an M/G/k with any service requirement distribution S which is either (i) bounded or (ii) unbounded with tail function of upper Matuszewska index*[8] *less than* -2,

$$\lim_{\rho\to 1}\frac{\mathbf{E}\left[T^{\text{PSJF-}k}\right]}{\mathbf{E}\left[T^{\text{SRPT-1}}\right]}=1.$$

Proof. From Theorem 4, we know that

$$T^{\text{PSJF-}k}(x)\le_{\text{st}} W^{\text{PSJF-1}}(x)+B_{\le x}((2k-1)x)$$

However, $W^{\text{PSJF-1}}(x)\le_{\text{st}} W^{\text{SRPT-1}}(x)$ [11]. Therefore,

$$\begin{aligned}T^{\text{PSJF-}k}(x)&\le_{\text{st}} W^{\text{SRPT-1}}(x)+B_{\le x}((2k-1)x)\\&\le_{\text{st}} W^{\text{SRPT-1}}(x)+B_{\le x}(2kx)\end{aligned}$$

This bound on $T^{\text{PSJF-}k}(x)$ is the same as the bound on $T^{\text{SRPT-}k}(x)$ given in Theorem 1. The rest of the proof proceeds as in the proof of Theorem 3. □

As in Corollary 1, Theorem 5 and the optimality of SRPT-1 imply that PSJF-k is optimal in the heavy-traffic limit.

7.2. Remaining size times original size (RS-k)

As usual, we consider a tagged job j of size x. When j has remaining size y, another job ℓ is *relevant* to j if the *product of* ℓ*'s original size and remaining size* is at most xy. In particular, if ℓ is relevant to j, then ℓ's remaining size is at most x. With this definition of relevance, we divide work into the same four categories as in Section 5, namely tagged, old, new, and virtual. This bounds the response time of j by

$$T^{\text{RS-}k}\le_{\text{st}} B_{\le x}\Big(x+\texttt{RelWork}^{\text{RS-}k}_{\le x}+\texttt{VirtWork}^{\text{RS-}k}(x)\Big). \tag{7.3}$$

The proof of Lemma 1 works nearly verbatim for RS-k, so

$$\texttt{VirtWork}^{\text{RS-}k}(x)\le(k-1)x. \tag{7.4}$$

The analysis of steady-state relevant work is similar to that in Section 5.2. We consider a pair of systems experiencing the same arrival sequence: System 1, which uses RS-1, and System k, which uses RS-k. We define $\Delta^{\text{RS-}k}_{\le x}(t)$ to be the difference between the amounts of relevant work in the two systems at time t. We then bound $\Delta^{\text{RS-}k}_{\le x}(t)$.

Lemma 6. *The difference in relevant work between Systems* 1 *and* k *is bounded by*

$$\Delta^{\text{RS-}k}_{\le x}(t)\le kx.$$

Proof. Even though RS uses a definition of relevant jobs different from SRPT's, the proof is analogous to that of Lemma 2. □

Theorem 6. *In an M/G/k, the response time of a job of size x under RS-k is bounded by*

$$T^{\text{RS-}k}(x)\le_{\text{st}} W^{\text{RS-1}}(x)+B_{\le x}((2k-1)x).$$

Proof. By (7.3), (7.4), and Lemma 6,

$$\begin{aligned}T^{\text{RS-}k}(x)&\le_{\text{st}} B_{\le x}\big(\texttt{RelWork}^{\text{RS-1}}_{\le x}+2kx\big)\\&\le_{\text{st}} B_{\le x}\big(\texttt{RelWork}^{\text{RS-1}}_{\le x}\big)+B_{\le x}(2kx).\end{aligned}$$

The waiting time in RS-1 is

$$W^{\text{RS-1}}(x)=B_{\le x}\big(\texttt{RelWork}^{\text{PSJF-1}}_{\le x}\big),$$

giving the desired bound. □

With the bound derived in Theorem 6, we can prove that RS-k also minimizes mean response time in the heavy-traffic limit.

Theorem 7. *In an M/G/k with any service requirement distribution S which is either (i) bounded or (ii) unbounded with tail function of upper Matuszewska index*[9] *less than* -2,

$$\lim_{\rho\to 1}\frac{\mathbf{E}\left[T^{\text{RS-}k}\right]}{\mathbf{E}\left[T^{\text{SRPT-1}}\right]}=1.$$

[8] See Section 2.1 or Appendix B.

Proof. From Theorem 6, we know that

$$T^{\text{RS-}k}(x) \leq_{\text{st}} W^{\text{RS-1}}(x) + B_{\leq x}(2kx)$$

However, $W^{\text{RS-1}}(x) \leq_{\text{st}} W^{\text{SRPT-1}}(x)$ [11]. Therefore,

$$T^{\text{RS-}k}(x) \leq_{\text{st}} W^{\text{SRPT-1}}(x) + B_{\leq x}(2kx)$$

This bound on $T^{\text{RS-}k}(x)$ is the same as the bound on $T^{\text{SRPT-}k}(x)$ given in Theorem 1. The rest of the proof proceeds as in the proof of Theorem 3. □

As in Corollary 1, Theorem 7 and the optimality of SRPT-1 imply that RS-k is optimal in the heavy-traffic limit.

We have so far shown response time bounds for SRPT-k, PSJF-k, and RS-k that are strong enough to prove asymptotic optimality in heavy traffic. We conjecture that similar bounds and optimality results hold for multiserver variants of any policy in the SMART class [11], which includes SRPT, PSJF, and RS.

7.3. Foreground–Background(FB-k)

The analysis of FB-k proceeds similarly to the analysis of SRPT-k but with a few more changes than were needed for PSJF-k and RS-k. To analyze PSJF-k and RS-k, we followed the same outline as Section 5 with a small change to the definition of relevant jobs. In particular, we reused the notion of relevant busy periods $B_{\leq x}(\cdot)$ from Definition 3. In contrast, as we will see shortly, FB-k has a significantly different definition of relevant jobs, so the definition of relevant busy periods will also change.

As usual, we consider a tagged job j of size x. Recall that FB prioritizes the jobs of smallest *age*, or attained service. When j arrives, its age is 0, so it has priority over all other jobs in the system. However, as j is served, its age increases and its priority gets worse. The key to the usual single-server analysis of FB is that to define relevant work, we have to look at j's *worst future priority* [31–33]. This worst priority occurs when j has age x, an instant before completion, giving us the following definition of relevant jobs.

Definition 5. Suppose job j has original size x. Under FB-k, a job ℓ is *relevant* to job j if ℓ has age at most x. Otherwise ℓ is *irrelevant* to j.

There is an important difference between the notions of relevance for SRPT-k and FB-k. Under SRPT-k, each arriving job starts as either relevant or irrelevant to j and remains that way for j's entire time in the system. In contrast, under FB-k, *every new arrival is at least temporarily relevant to* j. Specifically, if a new arrival ℓ has size at most x, then ℓ is relevant to j for its entire time in the system. If ℓ instead has size greater than x, then ℓ is relevant to j only until it reaches age x, at which point it becomes irrelevant. This observation motivates the definition of relevant busy periods for FB-k.

Definition 6. Under FB-k, a *relevant busy period* for a job of size x started by (possibly random) amount of work V, written $B_{\bar{x}}(V)$, is the amount of time it takes for a work-conserving system that starts with V work to become empty, where *every arrival's service is truncated at age* x. A relevant busy period has expectation

$$\mathbf{E}\left[B_{\bar{x}}(V)\right] = \frac{\mathbf{E}\left[V\right]}{1-\rho_{\bar{x}}}.$$

Above, $\rho_{\bar{x}}$ is the *relevant load* for a job of size x, which is the total load due to relevant jobs. Its value is

$$\rho_{\bar{x}} = \lambda \mathbf{E}\left[\min(S, x)\right],$$

because each arrival is relevant only until it reaches age x.

We make a similar modification to the definition of steady-state relevant work.

Definition 7. The *steady-state relevant work* for a job of size x under FB-k, written $\mathtt{RelWork}_{\bar{x}}^{\text{FB-}k}$, is the sum of *remaining truncated sizes* of all jobs observed at a random point in time. A job's remaining truncated size is the amount of time until it either completes or reaches age x.

Armed with Definitions 5–7, we divide work into the same four categories as in Section 5, namely tagged, old, new, and virtual. This bounds the response time of j by

$$T^{\text{FB-}k} \leq_{\text{st}} B_{\bar{x}}\Big(x + \mathtt{RelWork}_{\bar{x}}^{\text{FB-}k} + \mathtt{VirtWork}^{\text{FB-}k}(x)\Big). \tag{7.5}$$

The proof of Lemma 1 works nearly verbatim for FB-k, so

$$\mathtt{VirtWork}^{\text{FB-}k}(x) \leq (k-1)x. \tag{7.6}$$

[9] See Section 2.1 or Appendix B.

The analysis of steady-state relevant work is similar to that in Section 5.2. We consider a pair of systems experiencing the same arrival sequence: System 1, which uses FB-1, and System k, which uses FB-k. We define $\Delta_{\bar{x}}^{\text{FB-}k}(t)$ to be the difference between the amounts of relevant work in the two systems at time t. We then bound $\Delta_{\bar{x}}^{\text{FB-}k}(t)$.

Lemma 7. *The difference in relevant work between Systems* 1 *and* k *is bounded by*

$$\Delta_{\bar{x}}^{\text{FB-}k}(t) \le (k-1)x.$$

Proof. Even though FB uses a definition of relevant jobs different from PSJF's,[10] the proof is analogous to that of Lemma 5. □

Theorem 8. *In an M/G/k, the response time of a job of size x under FB-k is bounded by*

$$T^{\text{FB-}k}(x) \le_{\text{st}} B_{\bar{x}}\left(\texttt{RelWork}_{\bar{x}}^{\text{FB-}k} + (2k-1)x\right).$$

Proof. Combining (7.5), (7.6), and Lemma 7 yields the desired bound. □

Note that the waiting time under FB-1 is always zero, as a new job immediately receives service, so we do not phrase the bound in terms of waiting time.

With the bound derived in Theorem 6, we can prove that the mean response time of FB-k approaches that of FB-1 in the heavy-traffic limit. We make use of prior work on the mean response time of FB in heavy traffic [34]. Let

$$W(x) = \mathbf{E}\left[B_{\bar{x}}(\texttt{RelWork}_{\bar{x}}^{\text{FB-}k})\right]$$
$$R(x) = \mathbf{E}\left[B_{\bar{x}}(x)\right].$$

$W(x)$ and $R(x)$ are not the mean waiting and residence times of a job of size x under FB because waiting time is always zero, but they play roughly analogous roles in the standard analysis of FB [33, Section 5].

Lemma 8. *In an M/G/1 with any service requirement distribution S which is unbounded with tail function of upper Matuszewska index*[11] *less than* -2,

$$\lim_{\rho\to 1} \frac{\mathbf{E}\left[R(S)\right]}{\mathbf{E}\left[T^{\text{FB-}1}\right]} = 0.$$

Proof. Follows immediately from results of Kamphorst and Zwart [34]. See Appendix D. □

Theorem 9. *In an M/G/k with any service requirement distribution S which is unbounded with tail function of upper Matuszewska index less than* -2,

$$\lim_{\rho\to 1} \frac{\mathbf{E}\left[T^{\text{FB-}k}\right]}{\mathbf{E}\left[T^{\text{FB-}1}\right]} = 1.$$

Proof. The standard analysis of FB-1 [31,32] shows

$$\mathbf{E}\left[T^{\text{FB-}1}\right] = \mathbf{E}\left[W(S)\right] + \mathbf{E}\left[R(S)\right],$$

whereas Theorem 8 implies

$$\mathbf{E}\left[T^{\text{FB-}k}\right] \le \mathbf{E}\left[W(S)\right] + (2k-1)\mathbf{E}\left[R(S)\right],$$

so the result follows by Lemma 8. □

Righter and Shanthikumar [14] show that when the job size distribution S has decreasing hazard rate, FB-1 is optimal for minimizing response time among all scheduling policies that do not have access to job sizes. Theorem 9 implies that in the heavy-traffic limit, FB-k is optimal in the same setting.[12]

Corollary 3. *In an M/G/k with any service requirement distribution S which (a) is unbounded, (b) has decreasing hazard rate, and (c) has tail function of upper Matuszewska index less than* -2,

$$\lim_{\rho\to 1} \frac{\mathbf{E}\left[T^{\text{FB-}k}\right]}{\mathbf{E}\left[T^{P}\right]} \le 1$$

for any scheduling policy P that does not have access to job sizes.

[10] We draw an analogy with PSJF rather than SRPT because under both FB and PSJF, irrelevant jobs never become relevant.

[11] See Section 2.1 or Appendix B.

[12] It has been claimed that FB-k is optimal for arbitrary arrival sequences when the service requirement distribution has decreasing hazard rate [35, Theorem 2.1]. However, the proof has an error. See Appendix E.

7.4. What about first-come, first-served?

Having seen the success of our modified tagged job analysis for a variety of policies, it is natural to ask: does a similar analysis work for the multiserver First-Come, First-Served policy (FCFS-k)?

Unfortunately, our technique does not work for FCFS-k. To see why, let us take a look at what our analyses of SRPT-k, PSJF-k, RS-k, and FB-k have in common. A central component of all four analyses is bounding the *difference in relevant work* between two systems experiencing the same arrival sequence, one using a single-server policy P-1 and another using its k-server variant P-k. These bounds are given in Lemmas 2 and 5–7. All four lemmas have similar two-step proofs.

- First, they bound the *number of relevant jobs* both during few-jobs intervals and at the start of many-jobs intervals. For all four policies, this bound is at most k.
- Second, they bound the *relevant work contributed by each relevant job*. For all four policies, this bound is x.

When we try to prove analogous bounds for FCFS-k, we can still bound the number of relevant jobs by k, but *the relevant work contributed by each relevant job is unbounded*.

The definition of relevant jobs is the crucial difference between FCFS-k and the policies we analyze. Consider the jobs relevant to a tagged job j of size x.

- Under SRPT-k, PSJF-k, and RS-k, only *some* jobs are relevant to j, and all such jobs have size at most x.
- Under FB-k, while all jobs might be relevant to j, they are only *temporarily* relevant, each contributing at most x relevant work.
- However, under FCFS-k, *all* jobs in the system when j arrives are *permanently* relevant to j.

This means that if the service requirement distribution S is unbounded, our worst-case technique is insufficient for bounding the difference in relevant work between FCFS-1 and FCFS-k.

8. Conclusion

We give the first stochastic bound on the response time of SRPT-k (see Section 5). Using this bound, we show that SRPT-k has asymptotically optimal mean response time in the heavy-traffic limit (see Section 6). We generalize our analysis to give the first stochastic bounds on the response times of the PSJF-k, RS-k and FB-k policies, and we use these bounds to prove asymptotic optimality results for all three policies (see Section 7).

To achieve these results, we strategically combine stochastic and worst-case techniques. Specifically, we obtain our bounds using a modified tagged job analysis. Traditional tagged job analyses for single-server systems rely on properties that do not hold in multiserver systems, notably work conservation. To make tagged job analysis work for multiple servers, we use two key insights.

- We introduce the concept of *virtual work* (see Section 5), which makes the system appear work-conserving while the tagged job is in the system. We give a worst-case bound for virtual work.
- We compare the multiserver system with a *single-server system of the same service capacity*. We show that even in the worst case, the steady state amount of relevant work under SRPT-k is close to the steady state amount of relevant work under SRPT-1.

Applying these two insights to the tagged job analysis gives a *stochastic* expression bounding response time.

One direction for future work is to apply our technique to a broader range of scheduling policies. In particular, we conjecture that out results generalize to the SMART class of policies [11], which includes SRPT, PSJF, and RS. Another direction is to improve our response time bounds under low system load. While our bounds are valid for all loads, they are only tight for load near capacity.

Appendix A. Improved SRPT-*k* bound

Theorem 2. *In an M/G/k, the mean response time of a job of size x under SRPT-k is bounded by*

$$\mathbf{E}\left[T^{\text{SRPT-}k}(x)\right] \le \frac{\int_0^x \lambda t^2 f_S(t)\,dt}{2(1-\rho_{\le x})^2} + \frac{k\rho_{\le x}x}{1-\rho_{\le x}} + \int_0^x \frac{k}{1-\rho_{\le t}}\,dt,$$

where $f_S(\cdot)$ is the probability density function of the service requirement distribution S.

Proof. We will prove Theorem 2 by proving improved versions of (5.1) and Lemma 2.

A key element of our analysis is bounding the amount of new work done while the tagged job j of size x is in the system. In (5.1), we bound this quantity by a relevant busy period with size cutoff x. However, in reality, the size cutoff decreases as j receives service. We can use this to give a tighter bound on the amount of new work performed.

Let r_j be the amount of relevant work seen by j on arrival. Note that r_j is also the amount of old work that will be done while j is in the system.

Starting from the time of j's arrival, after at most $B_{\le x}(r_j)$ time, j must enter service. During this busy period, an amount of work is performed equal to r_j plus all relevant arrivals during this busy period.

More generally, for any amount of time $s \le x$, after at most a relevant busy period started by $r_j + ks$ work, j must have received s service. This holds because even if the servers finish all the old work and all the new work that has arrived so far, the servers must still complete ks combined tagged and virtual work. Of this tagged and virtual work, at least s must be tagged work, namely serving j. This means that the first dt service of j must be completed by time

$$B_{\le x}(r_j) + B_{\le x}(k \cdot dt).$$

The next dt service of j must be completed by time

$$B_{\le x}(r_j) + B_{\le x}(k \cdot dt) + B_{\le x-dt}(k \cdot dt),$$

because the cutoff for entering the relevant busy period decreases as j receives service. Similarly, the following dt service of j must be completed by time

$$B_{\le x}(r_j) + B_{\le x}(k \cdot dt) + B_{\le x-dt}(k \cdot dt) + B_{\le x-2\,dt}(k \cdot dt).$$

This pattern continues as j receives service. The descending size cutoff yields the same sort of relevant busy period as in the traditional tagged job analysis of SRPT-1 [6]. Recalling that r_j is drawn from the distribution $\texttt{RelWork}^{\text{SRPT-}k}_{\le x}$ yields the following bound on the mean response time of j:

$$T^{\text{SRPT-}k}(x) \le B_{\le x}\left(\texttt{RelWork}^{\text{SRPT-}k}_{\le x}\right) + \int_0^x B_{\le t}(k \cdot dt). \tag{A.1}$$

With (A.1), we have improved upon (5.1).

Next, we will improve upon Lemma 2. We consider a pair of systems experiencing the same arrival sequence: System 1, which uses PSJF-1, and System k, which uses SRPT-k.

Recall from Section 7.1 that under PSJF-1, a job ℓ is relevant to j if ℓ has *original* size at most x. In contrast, under SRPT-k, a job ℓ is relevant to j if ℓ has *remaining* size at most x.

We define $\Delta'_{\le x}(t)$ to be the difference between the amounts of relevant work in the two systems at time t. Using Lemma 9 (proof deferred), we obtain a bound on $\Delta'_{\le x}(t)$ tighter than the analogous bound in Lemma 2.

Lemma 9. *The difference in relevant work between Systems 1 and k is bounded by*

$$\Delta'_{\le x}(t) \le x \cdot \texttt{RelBusy}^{(k)}_{\le x}(t)$$

where $\texttt{RelBusy}^{(k)}_{\le x}(t)$ *is the number of servers in System k which are busy with relevant work at time t.*

Proof. We define few-jobs intervals and many-jobs intervals as in Section 5.2. Note that $\texttt{RelBusy}^{(k)}_{\le x}(t) = k$ during a many-jobs interval, and that $\texttt{RelBusy}^{(k)}_{\le x}(t)$ is the number of jobs in the system during a few-jobs interval.

The case where t is in a few-jobs interval is simple: there are exactly $\texttt{RelBusy}^{(k)}_{\le x}(t)$ jobs in System k at time t, each of remaining size at most x, so

$$\Delta'_{\le x}(t) \le x \cdot \texttt{RelBusy}^{(k)}_{\le x}(t).$$

Suppose instead that t is in a many-jobs interval, in which case $\texttt{RelBusy}^{(k)}_{\le x}(t) = k$. Let time s be the start of the many-jobs interval containing t. Over the interval $[s, t]$, the same amount of relevant work arrives in both systems, because relevant arrivals are the same under SRPT and PSJF. Upon arrival a job's original and remaining sizes are equal. The other two categories of relevant work over the interval follow the same arguments as in the proof of Lemma 2. Thus,

$$\Delta'_{\le x}(t) \le \Delta'_{\le x}(s).$$

It therefore suffices to show $\Delta'_{\le x}(s) \le kx$. As in Lemma 2, a many-jobs interval can begin due to the arrival of a relevant job, or due an irrelevant job in System k becoming relevant. In the case of an arrival, the same arrival occurs in System 1, and must be relevant in System 1, so

$$\Delta'_{\le x}(s) = \Delta_{\le x}(s^-) \le (k-1)x,$$

because s^-, the instant before s, is in a few-jobs interval. In the case of an irrelevant job in System k becoming relevant, by the same argument as in the proof of Lemma 2,

$$\Delta'_{\le x}(s) \le \texttt{RelWork}^{(k)}_{\le x}(s) \le kx. \quad \square$$

Continuing the proof of Theorem 2, we are now ready to prove the stronger bound. From (A.1), we know

$$T^{\text{SRPT-}k}(x) \le B_{\le x}\big(\texttt{RelWork}_{\le x}^{\text{SRPT-}k}\big) + \int_0^x B_{\le t}(k \cdot dt).$$

By plugging in Lemmas 1 and 9, we find that

$$\begin{aligned} &T^{\text{SRPT-}k}(x) \\ &\le B_{\le x}\big(\texttt{RelWork}_{\le x}^{\text{PSJF-1}} + x \cdot \texttt{RelBusy}_{\le x}^{\text{SRPT-}k}\big) + \int_0^x B_{\le t}(k \cdot dt) \\ &= B_{\le x}\big(\texttt{RelWork}_{\le x}^{\text{PSJF-1}}\big) + B_{\le x}\big(x \cdot \texttt{RelBusy}_{\le x}^{\text{SRPT-}k}\big) + \int_0^x B_{\le t}(k \cdot dt) \\ &= W^{\text{PSJF-1}}(x) + B_{\le x}\big(x \cdot \texttt{RelBusy}_{\le x}^{\text{SRPT-}k}\big) + \int_0^x B_{\le t}(k \cdot dt), \end{aligned}$$

where $\texttt{RelBusy}_{\le x}^{\text{SRPT-}k}$ is the steady state number of servers which are busy with relevant jobs under SRPT-k. Taking expectations yields

$$\mathbf{E}\left[T^{\text{SRPT-}k}(x)\right] \le \mathbf{E}\left[W^{\text{PSJF-1}}(x)\right] + \mathbf{E}\left[B_{\le x}\big(x \cdot \texttt{RelBusy}_{\le x}^{\text{SRPT-}k}\big)\right] + \int_0^x \mathbf{E}\left[B_{\le t}(k \cdot dt)\right].$$

From the literature [11], we know that

$$\mathbf{E}\left[W^{\text{PSJF-1}}(x)\right] = \frac{\int_0^x \lambda t^2 f_S(t)\, dt}{2(1-\rho_{\le x})^2}.$$

By the expectation of a relevant busy period, from Definition 3,

$$\int_0^x \mathbf{E}\left[B_{\le t}(k \cdot dt)\right] = \int_0^x \frac{k}{1-\rho_{\le t}}\, dt.$$

Similarly,

$$\mathbf{E}\left[B_{\le x}\big(x \cdot \texttt{RelBusy}_{\le x}^{\text{SRPT-}k}\big)\right] = \frac{\mathbf{E}\left[x \cdot \texttt{RelBusy}_{\le x}^{\text{SRPT-}k}\right]}{1-\rho_{\le x}}.$$

The average rate at which the SRPT-k system performs relevant work is $\mathbf{E}\left[\texttt{RelBusy}_{\le x}^{\text{SRPT-}k}\right]/k$, since each busy server does work at rate $1/k$. Because the system is stable, the rate at which relevant work is done must equal the rate at which relevant work enters the system, namely $\rho_{\le x}$. Thus, $\mathbf{E}\left[\texttt{RelBusy}_{\le x}^{\text{SRPT-}k}\right] = k\rho_{\le x}$, so

$$\mathbf{E}\left[B_{\le x}\big(x \cdot \texttt{RelBusy}_{\le x}^{\text{SRPT-}k}\big)\right] = \frac{k\rho_{\le x} x}{1-\rho_{\le x}},$$

yielding the desired bound. □

Appendix B. Matuszewska index

The heavy-traffic results in this paper, such as Theorem 3, assume that the service requirement distribution S is not too heavy-tailed. Specifically, we require that either S is bounded or that the *upper Matuszewska index* of the tail of S is less than -2. This is slightly stronger than assuming that S has finite variance. The formal definition of the upper Matuszewska index is the following.

Definition 8. Let f be a positive real function. The *upper Matuszewska index* of f, written $M(f)$, is the infimum over α such that there exists a constant C such that for all $\gamma > 1$,

$$\lim_{x\to\infty} \frac{f(\gamma x)}{f(x)} \le C\gamma^{\alpha}.$$

Moreover, for all $\Gamma > 1$, the convergence as $x \to \infty$ above must be uniform in $\gamma \in [1, \Gamma]$.

The condition $M(\bar{F}_S) < -2$, where $\bar{F}_S$ is the tail of S, is intuitively close to saying that $F_S(x) \le Cx^{-2-\epsilon}$ for some constant C and some $\epsilon > 0$. Roughly speaking, this means that S has a lighter tail than a Pareto distribution with $\alpha = 2$.

Appendix C. SRPT-1 in heavy traffic

Lemma 3. *In an M/G/1 with any service requirement distribution S which is either (i) bounded or (ii) unbounded with tail function of upper Matuszewska index*[13] *less than −2,*

$$\lim_{\rho\to 1}\frac{\log\left(\frac{1}{1-\rho}\right)}{\mathbf{E}\left[T^{\text{SRPT-1}}\right]}=0.$$

Proof. Lin et al. [21] show in their Theorem 1 that if S is bounded, then

$$\mathbf{E}\left[T^{\text{SRPT-1}}\right]=\Theta\left(\frac{1}{1-\rho}\right),$$

proving case (i) They also show in their Theorem 2 that if the upper Matuszewska index of the tail of S is less than −2, then

$$\mathbf{E}\left[T^{\text{SRPT-1}}\right]=\Theta\left(\frac{1}{(1-\rho)G^{-1}(\rho)}\right),$$

where $G^{-1}(\cdot)$ is the inverse of $G(x)=\rho_{\le x}/\rho$. In their proof of Theorem 2, they also show that

$$\lim_{\rho\to 1}\log\left(\frac{1}{1-\rho}\right)\cdot(1-\rho)G^{-1}(\rho)=0,$$

proving case (ii) □

Appendix D. FB-1 in heavy traffic

Lemma 8. *In an M/G/1 with any service requirement distribution S which is unbounded with tail function of upper Matuszewska index less than −2,*

$$\lim_{\rho\to 1}\frac{\mathbf{E}\left[R(S)\right]}{\mathbf{E}\left[T^{\text{FB-1}}\right]}=0.$$

Proof. Recall that

$$W(x)=\mathbf{E}\left[B_{\bar{x}}(\texttt{RelWork}_{\bar{x}}^{\text{FB-}k})\right]$$
$$R(x)=\mathbf{E}\left[B_{\bar{x}}(x)\right].$$

The standard analysis of FB-1 [31,32] shows

$$\mathbf{E}\left[T^{\text{FB-1}}\right]=\mathbf{E}\left[W(S)\right]+\mathbf{E}\left[R(S)\right].$$

Kamphorst and Zwart [34, Equation (4.3)] decompose $\mathbf{E}\left[T^{\text{FB-1}}\right]$ into a sum of three functions of the load ρ,

$$\mathbf{E}\left[T^{\text{FB-1}}\right]=X(\rho)+Y(\rho)+Z(\rho),$$

such that

$$\mathbf{E}\left[W(S)\right]=Z(\rho)+\frac{1}{2}Y(\rho)$$
$$\mathbf{E}\left[R(S)\right]=X(\rho)+\frac{1}{2}Y(\rho).$$

Kamphorst and Zwart [34, Section 4.1.1] then show that

$$\lim_{\rho\to 1}\frac{X(\rho)}{Z(\rho)}=\lim_{\rho\to 1}\frac{Y(\rho)}{Z(\rho)}=0,$$

which implies the desired limit

$$\lim_{\rho\to 1}\frac{\mathbf{E}\left[R(S)\right]}{\mathbf{E}\left[T^{\text{FB-1}}\right]}=\lim_{\rho\to 1}\frac{X(\rho)+\frac{1}{2}Y(\rho)}{X(\rho)+Y(\rho)+Z(\rho)}=0.\quad\square$$

[13] See Section 2.1 or Appendix B.

Appendix E. Flawed interchange arguments

Down and Wu [20, Theorem 2.1] claim that SRPT-k is optimal in the sense of minimizing the completion time of the nth job for all n, under the additional assumption that all servers are busy at all times. Unfortunately, this claim is false. The proof attempts to use an interchange argument, mimicking the classic proof of the optimality of SRPT-1 [7]. However, the specified interchange can result in the same job running on two servers simultaneously, which is of course not possible.

A concrete counterexample is the following: let $k = 2$, and let jobs of size 1, 1, 2 and 2 arrive at time 0. Recall that a job of size x must be in service for kx time to complete. SRPT-k completes its third job at time 6, while a policy which serves a job of size 2 over the interval [0, 4) and jobs of size 1 over the intervals [0, 2) and [2, 4) would finish its third job at time 4. Moreover, more complicated counterexamples exist which show that multiserver SRPT does not minimize mean response time even if all servers are busy at all times.

A similar error occurs in a claim by Wu and Down [35, Theorem 2.1] that FB-k is optimal among policies that do not have access to job size information when the service requirement distribution has decreasing hazard rate. Again the proof given is an interchange argument, and again the specified interchange can result in the same job running on two servers simultaneously.

References

[1] M. Harchol-Balter, B. Schroeder, N. Bansal, M. Agrawal, Size-based scheduling to improve web performance, ACM Trans. Comput. Syst. (ISSN: 0734-2071) 21 (2) (2003) 207–233, http://dx.doi.org/10.1145/762483.762486, http://doi.acm.org/10.1145/762483.762486.

[2] R. Mangharam, M. Demirhan, R. Rajkumar, D. Raychaudhuri, Size matters: Size-based scheduling for MPEG-4 over wireless channels, in: Multimedia Computing and Networking 2004, vol. 5305, International Society for Optics and Photonics, 2003, pp. 110–123.

[3] S. Guirguis, M.A. Sharaf, P.K. Chrysanthis, A. Labrinidis, K. Pruhs, Adaptive scheduling of web transactions, in: Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on, IEEE, 2009, pp. 357–368.

[4] R.B. Bunt, Scheduling techniques for operating systems, Computer 9 (10) (1976) 10–17.

[5] Y.H. Chen, P.A. Hsiung, Hardware task scheduling and placement in operating systems for dynamically reconfigurable SoC, in: L.T. Yang, M. Amamiya, Z. Liu, M. Guo, F.J. Rammig (Eds.), Embedded and Ubiquitous Computing – EUC 2005, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-32295-5, 2005, pp. 489–498.

[6] L.E. Schrage, L.W. Miller, The queue m/g/1 with the shortest remaining processing time discipline, Oper. Res. 14 (4) (1966) 670–684.

[7] L. Schrage, Letter to the editor-a proof of the optimality of the shortest remaining processing time discipline, Oper. Res. 16 (3) (1968) 687–690.

[8] S. Leonardi, D. Raz, Approximating total flow time on parallel machines, in: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, ACM, 1997, pp. 110–119.

[9] S. Leonardi, D. Raz, Approximating total flow time on parallel machines, J. Comput. System Sci. (ISSN: 0022-0000) 73 (6) (2007) 875–891, http://dx.doi.org/10.1016/j.jcss.2006.10.018, http://www.sciencedirect.com/science/article/pii/S0022000006001474.

[10] A. Wierman, M. Harchol-Balter, Classifying scheduling policies with respect to unfairness in an m/gi/1, in: ACM SIGMETRICS Performance Evaluation Review, ACM, 2003, pp. 238–249.

[11] A. Wierman, M. Harchol-Balter, T. Osogami, Nearly insensitive bounds on smart scheduling, in: ACM SIGMETRICS Performance Evaluation Review, ACM, 2005, pp. 205–216.

[12] E. Hyytiä, S. Aalto, A. Penttinen, Minimizing slowdown in heterogeneous size-aware dispatching systems, in: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, in: SIGMETRICS '12, ACM, New York, NY, USA, ISBN: 978-1-4503-1097-0, 2012, pp. 29–40, http://dx.doi.org/10.1145/2254756.2254763, http://doi.acm.org/10.1145/2254756.2254763.

[13] M. Nuyens, A. Wierman, The foreground–background queue: a survey, Perform. Eval. 65 (3–4) (2008) 286–307.

[14] R. Righter, J. Shanthikumar, Scheduling multiclass single server queueing systems to stochastically maximize the number of successful departures, Probab. Engrg. Inform. Sci. 3 (3) (1989) 323–333.

[15] S.C. Borst, O.J. Boxma, R. Nunez-Queija, Heavy tails: The effect of the service discipline, in: International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Springer, 2002, pp. 1–30.

[16] S.C. Borst, O.J. Boxma, R. Núñez-Queija, A. Zwart, The impact of the service discipline on delay asymptotics, Perform. Eval. 54 (2) (2003) 175–206.

[17] O. Boxma, B. Zwart, Tails in scheduling, SIGMETRICS Perform. Eval. Rev. (ISSN: 0163-5999) 34 (4) (2007) 13–20, http://dx.doi.org/10.1145/1243401.1243406, http://doi.acm.org/10.1145/1243401.1243406.

[18] N. Bansal, M. Harchol-Balter, Analysis of SRPT scheduling: Investigating unfairness, in: Proceedings of the 2001 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '01, ACM, New York, NY, USA, ISBN: 1-58113-334-0, 2001, pp. 279–290, http://dx.doi.org/10.1145/378420.378792, http://doi.acm.org/10.1145/378420.378792.

[19] M.E. Gebrehiwot, S. Aalto, P. Lassila, Energy-Aware server with SRPT scheduling: Analysis and optimization, in: G. Agha, B. Van Houdt (Eds.), Quantitative Evaluation of Systems, Springer International Publishing, Cham, ISBN: 978-3-319-43425-4, 2016, pp. 107–122.

[20] D.G. Down, R. Wu, Multi-layered round robin routing for parallel servers, Queueing Syst. 53 (4) (2006) 177–188.

[21] M. Lin, A. Wierman, B. Zwart, Heavy-traffic analysis of mean response time under shortest remaining processing time, Perform. Eval. (ISSN: 0166-5316) (2011) http://dx.doi.org/10.1016/j.peva.2011.06.001, http://www.sciencedirect.com/science/article/pii/S0166531611000721.

[22] N. Bansal, On the average sojourn time under M/M/1/SRPT, Oper. Res. Lett. 33 (2) (2005) 195–200.

[23] N. Bansal, D. Gamarnik, Handling load with less stress, Queueing Syst. 54 (1) (2006) 45–54.

[24] D.G. Down, H. Gromoll, A.L. Puha, Fluid limits for shortest remaining processing time queues, Math. Oper. Res. 34 (4) (2009) 880–911.

[25] I. Mitrani, P. King, Multiprocessor systems with preemptive priorities, Perform. Eval. (ISSN: 0166-5316) 1 (2) (1981) 118–125, http://dx.doi.org/10.1016/0166-5316(81)90014-6, http://www.sciencedirect.com/science/article/pii/0166531681900146.

[26] A. Sleptchenko, A.v. Harten, M.v.d. Heijden, An exact solution for the state probabilities of the multi-class, multi-server queue with preemptive priorities, Queueing Syst. (ISSN: 1572-9443) 50 (1) (2005) 81–107, http://dx.doi.org/10.1007/s11134-005-0359-y, https://doi.org/10.1007/s11134-005-0359-y.

[27] M. Harchol-Balter, T. Osogami, A. Scheller-Wolf, A. Wierman, Multi-Server queueing systems with multiple priority classes, Queueing Syst. (ISSN: 1572-9443) 51 (3) (2005) 331–360, http://dx.doi.org/10.1007/s11134-005-2898-7, https://doi.org/10.1007/s11134-005-2898-7.

[28] N. Avrahami, Y. Azar, Minimizing total flow time and total completion time with immediate dispatching, in: Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures, SPAA '03, ACM, New York, NY, USA, ISBN: 1-58113-661-7, 2003, pp. 11–18, http://dx.doi.org/10.1145/777412.777415, http://doi.acm.org/10.1145/777412.777415.

[29] M. Gong, C. Williamson, Simulation evaluation of hybrid SRPT scheduling policies, in: Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004. MASCOTS 2004. Proceedings. The IEEE Computer Society's 12th Annual International Symposium on, IEEE, 2004, pp. 355–363.

[30] R.W. Wolff, Poisson arrivals see time averages, Oper. Res. 30 (2) (1982) 223–231, http://dx.doi.org/10.1287/opre.30.2.223, https://doi.org/10.1287/opre.30.2.223.

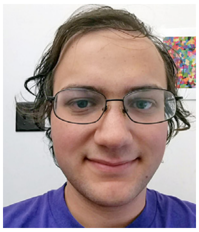
[31] M. Harchol-Balter, Performance modeling and design of computer systems: Queueing theory in action, Cambridge University Press, 2013.

[32] L.E. Schrage, The queue M/G/1 with feedback to lower priority queues, Manage. Sci. 13 (7) (1967) 466–474.

[33] Z. Scully, M. Harchol-Balter, A. Scheller-Wolf, SOAP: One clean analysis of all age-based scheduling policies, Proc. ACM Meas. Anal. Comput. Syst. (ISSN: 2476-1249) 2 (1) (2018) 16:1–16:30, http://dx.doi.org/10.1145/3179419, http://doi.acm.org/10.1145/3179419.

[34] B. Kamphorst, B. Zwart, Heavy-traffic analysis of sojourn time under the foreground-background scheduling policy, arXiv preprint arXiv:1712.03853, 2017..

[35] R. Wu, D.G. Down, Scheduling multi-server systems using foreground-background processing, in: The Forty-Second Allerton Conference, Citeseer, 2004.



Isaac Grosof received his B.S. and M.E. in Computer Science from the Massachusetts Institute of Computer Science in 2017. He is currently pursuing a Ph.D. in Computer Science at Carnegie Mellon University under Mor Harchol-Balter. His primary research interests are in queueing theory, scheduling theory and performance modeling.

Ziv Scully received his B.S. in Mathematics with Computer Science from the Massachusetts Institute of Technology in 2016. He is currently pursuing a Ph.D. in Computer Science at Carnegie Mellon University under Mor Harchol-Balter and Guy Blelloch, with primary research interests spanning queueing theory, performance analysis, and scheduling algorithms. His work has won finalist awards in the 2018 INFORMS APS Best Student Paper Competition and the 2016 PLDI Student Research Competition. In 2016 he received an NSF Graduate Research Fellowship and an ARCS Foundation scholarship.

Mor Harchol-Balter is a Professor of Computer Science at Carnegie Mellon. She received her Ph.D. from U.C. Berkeley in Computer Science in 1996 under the direction of Manuel Blum, and joined CMU in 1999. Mor is a Fellow of the ACM and a Senior Member of IEEE. She is a recipient of the McCandless Junior Chair, the NSF CAREER award, and several teaching awards, including the Herbert A. Simon Award. She has received faculty awards from a dozen companies including Google, Microsoft, IBM, EMC, Facebook, Intel, Yahoo!, and Seagate. Mor's work focuses on designing and analyzing new resource allocation policies, including load balancing policies, power management policies, and scheduling policies, for distributed systems. Mor is heavily involved in the SIGMETRICS/PERFORMANCE research community, where she has received multiple best paper awards, and where she served as TPC Chair in 2007, as General Chair in 2013, and as the Keynote Speaker for 2016. She is also the author of a popular textbook, *"Performance Analysis and Design of Computer Systems,"* published by Cambridge University Press, which bridges Operations Research and Computer Science. Mor is best known for her enthusiastic keynote talks and her many Ph.D. students, most of whom are professors in top academic institutions.