Analysis of SRPT Scheduling: Investigating Unfairness*

Nikhil Bansal and Mor Harchol-Balter
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{nikhil,harchol}@cs.cmu.edu

ABSTRACT

The Shortest-Remaining-Processing-Time (SRPT) scheduling policy has long been known to be optimal for minimizing mean response time (sojourn time). Despite this fact, SRPT scheduling is rarely used in practice. It is believed that the performance improvements of SRPT over other scheduling policies stem from the fact that SRPT unfairly penalizes the large jobs in order to help the small jobs. This belief has led people to instead adopt "fair" scheduling policies such as Processor-Sharing (PS), which produces the same expected slowdown for jobs of all sizes.

This paper investigates formally the problem of unfairness in SRPT scheduling as compared with PS scheduling. The analysis assumes an M/G/1 model, and emphasizes job size distributions with a heavy-tailed property, as are characteristic of empirical workloads. The analysis shows that the degree of unfairness under SRPT is surprisingly small.

The M/G/1/SRPT and M/G/1/PS queues are also analyzed under overload and closed-form expressions for mean response time as a function of job size are proved in this setting.

1. INTRODUCTION

It has been long known that always giving service to the job with the shortest-remaining-processing-time (SRPT) is the optimal scheduling policy with respect to minimizing the mean response time. Yet, many existing schedulers time-share the processor equally among all jobs, giving each job an equal quantum of service. For example, a web-server today time shares between its many concurrent open connections, giving each an approximately equal share of processing time. In the limit, as the size of the quantum goes to zero, this "fair-share" scheduling policy is known as Processor Sharing (PS).

There are reasons why the optimal policy SRPT is not prevalent in practice. In some cases, it is because the *size of a job* (its processing requirement) is not known in advance, so SRPT cannot be applied. However, in several applications this is not the case, and it is possible to reasonably estimate the size of a job. For example, in the case of static web requests to a web server, a job's processing requirement is proportional to the file size requested, which is known by the server. Likewise in several database applications, the processing requirement for a query may be estimated in advance.

A second objection to switching to SRPT is that it is not clear whether the performance improvements of SRPT over traditional scheduling policies like PS are significant. Comparing SRPT with other policies is not easy given the complex nature of existing performance formulas for SRPT.

However, the foremost and very commonly cited objection to using SRPT is the fear that large jobs may "starve" under SRPT [1, 28, 29, 26]. It is often stated that the huge average performance improvements of SRPT over other scheduling policies stem from the fact that SRPT unfairly penalizes the large jobs in order to help the small jobs. It is often thought that the performance of small jobs cannot be improved without hurting the large jobs (see Section 2) and thus large jobs suffer unfairly under SRPT.

This paper will investigate the objections cited above. Before we can state our results, we need to define the performance metrics and the workloads which we use. The performance metrics we use throughout are *response time* and *slowdown*. The response time of a job (a.k.a. sojourn time, turnaround time, flow time) is the time from when the job first arrives at the system until it departs the system. The slowdown of a job (a.k.a. stretch, normalized response time) is the ratio of its response time to its size. The slowdown metric is important because it helps to evaluate unfairness. For example, in an M/G/1 system with PS scheduling, all jobs (long and short) experience the same expected slowdown (hence PS is "fair").

It turns out that the job size distribution is important with respect to evaluating SRPT. We will therefore assume a general job size distribution. We will also concentrate on the special case of distributions with the **heavy-tailed prop-**

*This research was supported by Cisco Systems via a grant from the Pittsburgh Digital Greenhouse 00-1 and by NSF ITR 99-167 ANI-0081396

erty (**HT property**), where the largest 1% of the jobs comprise more than half the load. This HT property appears in many recent measurements of computing systems (see Section 3).

Throughout this paper we assume an M/G/1 queue where G is assumed to be a *continuous distribution* with *finite* mean and variance (the abbreviation *c.f.m.f.v.* is used to denote continuous, finite mean, finite variance in the theorems).

In the case where $\rho < 1$ we prove the following results:

On the topic of mean performance improvements:

- Although it is well-known that SRPT scheduling optimizes mean response time, it is not known how SRPT compares with PS with respect to mean slowdown. We prove that SRPT scheduling also outperforms PS scheduling with respect to mean slowdown for all job size distributions (Theorem 1, Section 4).
- Given that SRPT improves performance over PS both with respect to mean response time and mean slowdown, we next investigate the magnitude of the improvement. We prove that for all job size distributions with the HT property the improvement is very significant under high loads. For example, for load 0.9, SRPT improves over PS with respect to mean slowdown by a factor of at least 4 for all distributions with the HT property. As the load approaches 1, we find that SRPT improves over PS with respect to mean slowdown by a factor of 100 for all distributions with the HT property (Theorem 2, Section 4). In general we prove that for *all* job size distributions as the load approaches one, the mean response time under SRPT improves upon the mean response time under PS by at least a factor of 2 and likewise for mean slowdown. (Corollaries 1 and 2, Section 4).

On the topic of starvation we first show some counter-intuitive results:

- The performance improvement of SRPT over PS does *not* usually come at the expense of the large jobs (Section 5.1, Claim 1). In fact, we observe via example that for many job size distributions with the HT property every single job, including a job of the maximum possible size, prefers SRPT to PS (unless the load is extremely close to 1).
- While the above result does not hold at all loads, we prove that no matter what the load, at least 99% of the jobs have a lower expected response time under SRPT than under PS, for all job size distributions with the HT property (Section 5.2, Corollary 4). In fact, these 99% of the jobs do significantly better. We show that these jobs have an average slowdown of at most 4, at any load $\rho < 1$ (Section 5.2, Theorem 7), whereas their performance could be arbitrarily bad under PS as the load approaches 1. Similar, but weaker results are shown for general distributions (Section 5.2, Theorem 4 and 5).
- While the previous result is concerned only with 99% of the jobs, we also prove upper bounds on how much worse any job could fare under SRPT as opposed to PS for general distributions (Section 5.2, Theorem 6). Our bounds show that jobs never do too much worse under SRPT than under PS. For example, for all job size distributions, the expected response time under SRPT for any job is never more than 3 times that under PS, when the load is 0.8, and never more than 5.5 times that under PS when the load is 0.9. In fact, if the load is less than half, then for every job size distribution, each job has a lower expected response time and slowdown under SRPT than under PS (Section 5.2, Theorem 4).
- The above results show an upper bound on how much worse a job could fare under SRPT as opposed to PS for general job size distributions. We likewise prove lower bounds on the performance of SRPT as compared with PS for general job size distributions. (Section 5.2, Theorem 8).

Finally in the case where load $\rho > 1$ we prove that:

- Consider a job of size x such that $\rho(x) < 1$, where $\rho(x)$ denotes the load made up of jobs of size $\leq x$. For such jobs, we prove that the expected response time and slowdown are *finite* under SRPT. We derive a closed-form expression for the mean response time of a job of size x where $\rho(x) < 1$ under SRPT (see Section 7, Theorem 9). By contrast, under PS scheduling, it is well known that *all jobs*, including the very small ones experience infinite expected response time and slowdown for $\rho > 1$ [14].
- We evaluate our overload formula above, for the case of a heavy-tailed job size distribution. We show that for heavy-tailed job size distributions, for a system with average load well-above one, the mean response time for all but the largest 1% of the jobs is surprisingly low under SRPT. For example, under a load of $\rho = 1.5$, 99% of jobs will experience a mean slowdown of only 4 under SRPT scheduling, as compared with a mean slowdown of infinity for every job under PS scheduling (see Section 7, Figure 2).

There is certainly more work to be done on the problem of comparing SRPT versus PS scheduling under overload. Jean-Marie and Robert [11] provide some nice analysis of PS under overloaded conditions.

Throughout this paper, for the sake of clarity, we compare SRPT with PS scheduling only. The reason for this is that PS has the properties that it is (1) "ultimately" fair (equal slowdown for all jobs), (2) insensitive to the variance of the job size distribution, which implies good performance, and (3) ubiquitous. For completeness in Section 9 we also compare SRPT to other scheduling policies in the literature such as: first-come-first-server(FCFS), random (RANDOM), non-preemptive last-come-first-serve (LCFS), shortest-job-first (SJF), preemptive-last-come-first-served (P-LCFS) and feedback (FB) scheduling.

This paper argues why SRPT scheduling makes sense on a performance level. In practice, it is not always so obvious how SRPT scheduling should be applied, given that most systems have *multiple* devices and multiprogramming is necessary to ensure that cycles aren't wasted. For an example of SRPT being applied successfully to Web servers see [6].

2. PREVIOUS WORK

Mean results: It has long been known that SRPT has the lowest mean response time of any scheduling policy, given any arrival sequence and job sizes [23, 27]. Rajaraman et al. showed further that the mean slowdown under SRPT is at most twice the optimal mean slowdown for any sequence of job arrivals [5].

Schrage and Miller first derived the expressions for the response times in an M/G/1/SRPT queue [24]. This was further generalized by Pechinkin *et al.* to disciplines where the remaining times are divided into intervals. The jobs with remaining times in the smaller interval are served first but those within the same interval are served in first-come-first server order [18]. The steady-state appearance of the M/G/1/SRPT queue was obtained by Schassberger [22].

Though the above formulas have been known for a long time, they are difficult to evaluate numerically, due to their complex form (many nested integrals). Hence, the comparison of SRPT to other policies was long neglected. More recently, SRPT has been compared with other policies by plotting the mean response times for specific job size distributions under specific loads [21, 19, 25, 24, 7]. A 7-year long study at University of Aachen under Schreiber [19, 25] involved extensive evaluation of SRPT for various job size distributions and loads. The survey paper by Schreiber [25] summarizes the results. These results are all plots for *specific* job size distributions and loads. Hence it is not clear whether the conclusions based on these plots hold for more general job size distributions and loads.

Unfairness results: It has often been cited that SRPT may lead to *starvation* of large jobs [1, 28, 29, 26]. Usually, examples of adversarial arrival sequences where a particular job starves are given to justify this. However, such worst case examples do not reflect the behavior of SRPT in the average case.

The term "starvation" is also used by people to indicate the *unfairness* of SRPT's treatment of long jobs. It is often thought that since SRPT favors small jobs, long jobs should have a worse average performance under SRPT than under other policies. The argument given is that if a scheduling policy manages to reduce the response time of small jobs, then the response times for the large jobs would have to increase considerably. This argument does hold for scheduling policies which do not make use of size, see the famous Kleinrock Conservation Law [13], [14, Page 197]. However the argument does *not* necessarily apply to policies which make use of size, for example SRPT.

Very little has been done to evaluate the problem of unfairness analytically. Recently, Bender et al. consider the metric *max slowdown* of a job, as indication of unfairness [1]. They show with an example that SRPT can have an arbitrarily large *max slowdown*. However, *max slowdown* is not an appropriate metric to measure unfairness. A large job may have an exceptionally long response time in some case, but it might do well most of the time. A more relevant metric which we use in our paper is the *max mean slowdown*.

There has also been work in the area of proposing new SRPT-like policies [2, 17] which try to reduce the problem of unfairness, while still favoring the short jobs. These usually prioritize based on *both* the time a job has waited so far, and its remaining size. These policies are usually analytically intractable and have been evaluated by simulation only. However simulations show that they are promising.

Overload results: No formulas have been derived for M/G/1/SRPT under overload. In our derivation we use a combination of ideas from [24] and [12].

3. THE HEAVY-TAILED PROPERTY

Many application environments show a mixture of job sizes spanning many orders of magnitude. Much previous work has used the *exponential* distribution to capture this variability, However, recent measurements indicate that for many applications the exponential distribution is a poor model and that a *heavy-tailed* distribution is more accurate. In general a heavy-tailed distribution is one for which

$$Pr\{X > x\} \sim x^{-\alpha}, \qquad \text{where } 0 < \alpha < 2.$$

In practice, there is some maximum and minimum job size (forced by finite limits in system resources). Therefore, job sizes are often modeled as being generated i.i.d from a distribution that has a heavy-tailed form, but has finite upper and lower bounds. This truncated distribution is referred to as the *Bounded-Pareto* distribution [8]. It is characterized by three parameters: α, the exponent of the power law; k, the shortest possible job; and p, the largest possible job, The probability density function for the Bounded Pareto $B(k, p, \alpha)$ is defined as:

$$f(x) = \frac{\alpha k^\alpha}{1 - (k/p)^\alpha} x^{-\alpha - 1} \quad k \le x \le p, \quad 0 < \alpha < 2$$

Throughout this paper, whenever the $B(k, p, \alpha)$ distribution is mentioned, it will be assumed that k is chosen such that the mean value is fixed at 3000 and the maximum value fixed at $p = 10^{10}$, which correspond to typical values taken from [4].

Many recent measurements of computing systems [15, 9, 4, 10, 20] have observed job size distributions which are well-modeled by a Bounded Pareto distribution, where $\alpha \approx 1$.

One key property of heavy-tailed distributions and (many) Bounded Pareto distributions is that a tiny fraction ($< 1\%$) of the very largest jobs comprise over half of the total load. We will refer to this as the **heavy-tailed property (HT property)** *throughout this paper.* Observe that for lower values of α, the HT property is more pronounced, whereas it is less pronounced for higher values of α. Throughout the paper, whenever we use the Bounded Pareto distribution in the paper, it will always be the $BP(k = 332, p = 10^{10}, \alpha = 1.1)$ distribution. This distribution has a strong heavy-tailed property (the largest .3% of the jobs comprise half the total load), mean 3000, and variance $7.25 \cdot 10^{11}$. Note that while the Bounded Pareto distribution has both the HT property and finite moments, in general heavy-tailed distributions have the HT property, but infinite variance, and sometimes even infinite mean.

4. MEAN ANALYSIS OF M/G/1/SRPT

This section presents a comparison of the M/G/1/SRPT queue and the M/G/1/PS queue with respect to mean response time and mean slowdown.

We denote the average arrival rate by λ. We will assume that the job size distribution is c.f.m.f.v. with probability density function $f(t)$. The cumulative job size distribution will be denoted by $F(t)$. We will denote $1 - F(t)$ by $\bar{F}(t)$. X will refer to the service time of a job. The load (utilization), ρ, of the server is $\rho = \lambda \int_0^\infty t f(t) dt$. The load made up by the jobs of size less than or equal to x, $\rho(x)$, is $\rho(x) = \lambda \int_0^x t f(t) dt$. Let $m_2(x)$ be defined as follows: $m_2(x) = \int_0^x t^2 f(t) dt$.

The expected response time for a job of size x under SRPT, $E[T(x)]_{SRPT}$, can be decomposed into the expected waiting time of the job, $E[W(x)]_{SRPT}$, and the expected residence time of the job $E[R(x)]_{SRPT}$, where $E[W(x)]$ is the expected time for a job of size x from when it first arrives to when it receives service for the first time, and $E[R(x)]_{SRPT}$ is the expected residence time (the time it takes for a job of size x to complete once it begins execution). The formulas for these expressions are given by [24]

$$E[T(x)]_{SRPT} = E[W(x)]_{SRPT} + E[R(x)]_{SRPT} \quad (1)$$

$$E[W(x)]_{SRPT} = \frac{\lambda(m_2(x) + x^2(1 - F(x)))}{2(1 - \rho(x))^2} \quad (2)$$

$$E[R(x)]_{SRPT} = \int_0^x \frac{dt}{1 - \rho(t)} \quad (3)$$

For PS the expected response time for a job of size x, $E[T(x)]_{PS}$, is given by [30]

$$E[T(x)]_{PS} = \frac{x}{1 - \rho} \quad (4)$$

For any policy, if $E[T(x)]$ is the expected response time for a job of size x, then the expected *slowdown* for a job of size x, $E[S(x)]$, is given by

$$E[S(x)] = \frac{E[T(x)]}{x}$$

The mean response time and mean slowdown are given by $E[T] = \int_0^\infty E[T(x)] f(x) dx$ and $E[S] = \int_0^\infty E[S(x)] f(x) dx$ respectively.

Observe that for a given load ρ, all jobs have the same slowdown under PS, since, $E[S(x)]_{PS} = \frac{1}{1-\rho}$ for any x. Thus PS is ultimately "fair".

We now show that the mean performance advantages of SRPT over PS are significant.

THEOREM 1. *For load $\rho < 1$, for any c.f.m.f.v. distribution of job sizes,*

$$E[T]_{SRPT} \le h(\rho) E[T]_{PS}$$

$$E[S]_{SRPT} \le h(\rho) E[S]_{PS}$$

where

$$h(\rho) = \frac{\rho}{2} - \frac{(1 - \rho)\log(1 - \rho)}{\rho}$$

In particular, for any load ρ, $E[S]_{SRPT} \le E[S]_{PS}$.

The proof of Theorem 1 will be given in Section 6, since it requires analysis not yet developed.

Observing that $h(\rho) \to \frac{1}{2}$, as $\rho \to 1$, we get:

COROLLARY 1. *For any c.f.m.f.v. job size distribution, as the load $\rho \to 1$, $E[T]_{SRPT} \le \frac{1}{2} E[T]_{PS}$.*

COROLLARY 2. *For any c.f.m.f.v. job size distribution, as the load $\rho \to 1$, $E[S]_{SRPT} \le \frac{1}{2} E[S]_{PS}$.*

It is easy to see that the factor of two improvement in Corollaries 1 and 2 is in fact tight, given the assumption of general distributions. To see this, observe that for the constant job size distribution, SRPT is identical to FCFS. As the load approaches 1, it can be seen that $E[T]_{SRPT} = E[T]_{FCFS} \approx \frac{1}{2} E[T]_{PS}$.

The bound proven in Theorem 1 can be greatly strengthened if we limit our attention to job size distributions with the HT property.

THEOREM 2. *For load $\rho < 1$, for any c.f.m.f.v. job size distribution with the HT property,*

$$E[S]_{SRPT} \le k(\rho) E[S]_{PS}$$

where

$$k(\rho) = \frac{\rho(1.01 - \rho)}{2 - \rho} - \frac{2(1 - \rho)}{\rho} \left(\log(1 - \frac{\rho}{2}) + 0.01 \log \frac{1 - \rho}{1 - \frac{\rho}{2}} \right)$$

COROLLARY 3. *For any c.f.m.f.v. job size distribution with the HT property, as the load $\rho \to 1$,*

$$E[S]_{SRPT} \le \frac{1}{100} E[S]_{PS}$$

The proof of Theorem 2 will be given in Section 6, since it requires analysis not yet developed.

At this point it is tempting to assume that the large mean slowdown improvements of SRPT claimed above are due to disproportionately helping the many small jobs and sacrificing the fewer big jobs. In the next section we will show that this is in fact not the case.

5. UNFAIRNESS ANALYSIS

It is commonly believed that it is not possible to improve the performance of some jobs without hurting the performance of some other jobs. In section 5.1 we dispel this notion. We show with an example that there exist job size distributions such that *every* job can do better under SRPT than under PS. We also give intuition as to why this might be true. We then show the main analytical results on unfairness in Section 5.2.

5.1 All jobs can do better

We saw in Theorem 2 that, for job size distributions with the HT property, mean slowdown is substantially lower under SRPT as compared with PS. We now ask whether this *mean* improvement comes at the cost of severely penalizing large jobs. We now show that for at least one particular job size distribution, $BP(k, p, \alpha = 1.1)$, there is zero penalty to large jobs. Figure 1 below shows the slowdown as function of job size, at load 0.9 for the $BP(k, p, \alpha = 1.1)$. The plot shows the expected slowdown for a job in each percentile of the job size distribution (where 100 percentile indicates the very largest job, i.e., a job of size p.). Observe that, each job has an expected slowdown of 10 under PS, yet every single job has a smaller slowdown (and hence response time) under SRPT. Even the largest job has a slowdown of only 9.54 under SRPT. We state this observation as a claim, which is supported by Figure 1.

Claim 1. *There exist job size distributions such that every job does better under SRPT than under PS.*

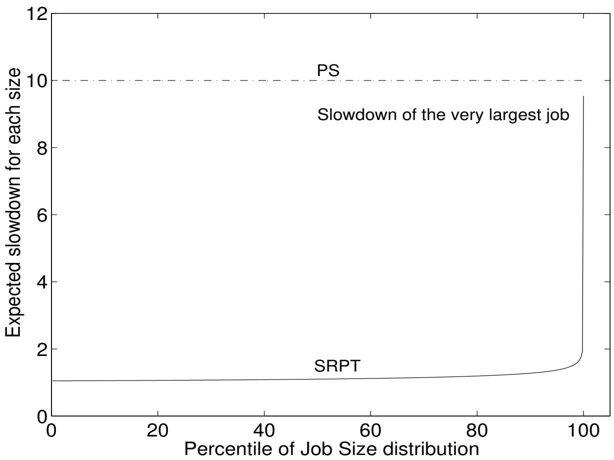

Figure 1: *Expected slowdown as a function of job size for $B(k, p, \alpha = 1.1)$ distribution, at load $\rho = 0.9$. Even a job of maximum size p prefers SRPT to PS.*

The above claim applies to many distributions with the HT property, provided ρ is not extremely close to 1. For example, if the job size distribution is $B(k, p, \alpha = 1.1)$, then every job does better under SRPT as long as the load is below 0.96. In fact for $B(k, p, \alpha = 1.5)$ every job does better for loads up to 0.999.

It is clear that SRPT should benefit the small jobs. However, it is not at all clear why SRPT should also benefit the large jobs. Intuitively, the following explains why this should be true:

Under SRPT, a job is affected only by the other jobs in the system which have a smaller remaining size than itself. Once a job begins execution, its remaining size diminishes with time. Thus the load seen by the job gets smaller as the job is worked upon. In contrast under PS, throughout its execution, a job is affected by *all* the other jobs present in the system. Thus the load that the job sees does not change with time. Thus it makes sense that, the expected *residence time* of a job under SRPT is smaller than its expected response time under PS. This difference is especially significant for distributions with the HT property, where the large jobs make up most of the load. To argue about *response time* under SRPT, however, we also need to take into account the *waiting time* under SRPT. Although the waiting time under SRPT may be large for big jobs, it turns out that provided the load isn't too high, the response time is dominated by residence time, not waiting time. In the next section, we will provide formal proofs which take all these details into account.

5.2 Unfairness analysis for general job size distributions

Theorem 1 shows the existence of job size distributions for which every job prefers SRPT to PS under most loads. We now extend this result along many directions. First, we show a similar but weaker result that holds for all c.f.m.f.v. job size distributions.

Theorem 3. *For any c.f.m.f.v. job size distribution, if the load is not more than half then every job has a lower expected response time under SRPT, as compared with PS.*

The proof of this theorem will follow from Theorem 4. The condition that the load is lower than half in Theorem 3 is rather restrictive. However, if we relax the restriction that *every* job performs better, then we get the following stronger result which holds at all loads.

Theorem 4. *For any c.f.m.f.v. job size distribution f and any load $\rho < 1$,*

$$E[T(x)]_{SRPT} \leq E[T(x)]_{PS}$$

for every job of size x such that $\rho(x) \leq \frac{1}{2}$ (i.e. jobs of size $\leq x$ comprise less than half the load).

Theorem 4 implies Theorem 3, since $\rho \leq \frac{1}{2}$ directly implies $\rho(x) \leq \frac{1}{2}$ for all x. The proof of Theorem 4 will follow from a more general Theorem 5 below.

Theorem 4 becomes especially useful if we relate the load percentiles and the job percentiles (i.e. $\rho(x)$ and $F(x)$). The HT property stated in Section 3 implies that less than 1% of the very largest jobs make up more than half the load. Thus Theorem 4 implies that at least 99% of the jobs have smaller response times under SRPT than under PS no matter what the load. Thus we have Corollary 4.

COROLLARY 4. *For c.f.m.f.v. distributions with the HT property, at least 99% of the jobs have a lower response time under SRPT than under PS at any load.*

Observe however that even for the "light-tailed" exponential ($C^2 = 1$) Theorem 4 implies that more than 81% of the jobs do better at any load, $\rho < 1$, under SRPT as compared with PS.

We now state and prove a generalization of Theorem 4 which likewise holds for any job size distribution and load $\rho < 1$.

THEOREM 5. *For any c.f.m.f.v. job size distribution f and load $\rho < 1$,*

$$E[T(x)]_{SRPT} \leq E[T(x)]_{PS}$$

for all jobs of size x such that

$$2(1-\rho(x))^2 \geq (1-\rho) \tag{5}$$

PROOF. $E[T(x)]_{PS}$ can be written as $\int_0^x \frac{dt}{1-\rho}$. Also, $E[R(x)]_{SRPT} = \int_0^x \frac{dt}{1-\rho(t)}$.
Since $\rho \geq \rho(t)$ for any t, the expected residence time for any job under SRPT is smaller than the expected response time under PS. We will bound this difference and obtain conditions under which the difference more than compensates for the waiting time under SRPT.

$$\begin{aligned}
& E[T(x)]_{PS} - E[R(x)]_{SRPT} \\
= & \int_0^x \frac{dt}{1-\rho} - \int_0^x \frac{dt}{1-\rho(t)} \\
= & \int_0^x \frac{(\rho-\rho(t))dt}{(1-\rho(t))(1-\rho)} \\
\geq & \int_0^x \frac{(\rho-\rho(t))dt}{1-\rho} \qquad [\text{Since} \quad (1-\rho(t)) \leq 1] \\
= & \frac{x(\rho-\rho(x)) + \lambda m_2(x)}{1-\rho} \qquad (6) \\
& [\text{Since} \quad \frac{d\rho(t)}{dt} = \lambda t f(t)] \\
\geq & \frac{\lambda x^2(1-F(x)) + \lambda m_2(x)}{1-\rho} \qquad (7)
\end{aligned}$$

Line (7) follows from Line (6) since:

$$\begin{aligned}
x(\rho-\rho(x)) &= \lambda x \int_x^\infty t f(t)dt \\
&\geq \lambda x^2 \int_x^\infty f(t)dt \\
&= \lambda x^2(1-F(x))
\end{aligned}$$

Comparing the expression for $E[W(x)]_{SRPT}$ in equation (2) with (7) it is clear that,

$$E[T(x)]_{PS} - E[R(x)]_{SRPT} \geq E[W(x)]_{SRPT}$$

whenever the condition (5) is met.
Thus, $E[T(x)]_{PS} \geq E[T(x)]_{SRPT}$ if $2(1-\rho(x))^2 \geq (1-\rho)$. ☐

PROOF. (**Theorem 4**) If $\rho(x) \leq \frac{1}{2}$, then $2(1-\rho(x)) \geq 1$. Observe that for all x, $(1-\rho(x)) \geq (1-\rho)$. Multiplying both the inequalities we get, $2(1-\rho(x))^2 \geq (1-\rho)$ and the result follows from Theorem 5. ☐
Theorems 3 and 4 show that for all job size distributions,

1. If $\rho \leq \frac{1}{2}$, then all jobs have a lower expected response time under SRPT as compared to PS.
2. Even if $\rho > \frac{1}{2}$, a majority of the jobs have better expected response times under SRPT.

But what about the small fraction of jobs which have a higher slowdown under SRPT than under PS, how bad can their starvation be? We will show that for a fixed load, no job can do arbitrarily badly on the average. Theorem 6 establishes an bound on the ratio of the expected response time of a job of size x under SRPT as compared with PS.

THEOREM 6. *For all c.f.m.f.v. job size distributions f, for all loads $\rho < 1$, for all x,*

$$E[T(x)]_{SRPT} \leq \frac{1-\rho}{1-\rho(x)}\left[\frac{\rho}{2(1-\rho(x))}+1\right] \cdot E[T(x)]_{PS} \tag{8}$$

In particular,

$$E[T(x)]_{SRPT} \leq \left[\frac{\rho}{2(1-\rho)}+1\right] \cdot E[T(x)]_{PS} \tag{9}$$

Before we can prove this theorem, we need one observation:

LEMMA 1.

$$\int_0^x tf(t)dt + x \cdot \bar{F}(x) \leq E[X]$$

PROOF.

$$E[X] = \int_0^x tf(t)dt + \int_x^\infty tf(t)dt \geq \int_0^x tf(t)dt + x\bar{F}(x)$$

☐

PROOF. (**Theorem 6**)

$$\begin{aligned}
E[T(x)]_{SRPT} &= \frac{\lambda \int_0^x t^2 f(t)dt + \lambda x^2 \bar{F}(x)}{2(1-\rho(x))^2} + \int_0^x \frac{dt}{1-\rho(t)} \\
&\leq \frac{\lambda \int_0^x t^2 f(t)dt + \lambda x^2 \bar{F}(x)}{2(1-\rho(x))^2} + \frac{x}{1-\rho(x)} \\
& \quad [\text{Since } (1-\rho(x)) \leq (1-\rho(t)), \text{for } t \leq x] \\
&\leq \frac{\lambda x \int_0^x tf(t)dt + \lambda x^2 \bar{F}(x)}{2(1-\rho(x))^2} + \frac{x}{1-\rho(x)} \\
&= \frac{x}{1-\rho(x)}\left[\frac{\lambda \int_0^x tf(t)dt + \lambda x \bar{F}(x)}{2(1-\rho(x))}+1\right] \\
&\leq \frac{x}{1-\rho(x)}\left[\frac{\lambda E[X]}{2(1-\rho(x))}+1\right] \\
& \quad [\text{By Lemma 1}] \\
&= \frac{x}{1-\rho}\frac{1-\rho}{1-\rho(x)}\left[\frac{\rho}{2(1-\rho(x))}+1\right] \\
&= E[T(x)]_{PS}\frac{1-\rho}{1-\rho(x)}\left[\frac{\rho}{2(1-\rho(x))}+1\right]
\end{aligned}$$

Thus equation (8) follows.

We observe that the expression $\frac{1-\rho}{1-\rho(x)}\left[\frac{\rho}{2(1-\rho(x))}+1\right]$ is maximized when x is the largest job (i.e. $\rho(x)=\rho$), in which case we get $E[T(x)]_{SRPT} \leq (\frac{\rho}{2(1-\rho)}+1)E[T(x)]_{PS}$. $\Box$

Theorem 6 shows that for a given a load ρ, the expected response time for a job cannot be arbitrarily worse under SRPT, as compared with PS. For example, if $\rho = 0.8$, the expected response time of every job under SRPT is no more than 3 times that under PS, and no more than 5.5 times that under PS where $\rho = 0.9$. In reality however, the factor is much better, since our analysis is not tight and it holds for all job size distributions. Stronger results can be obtained for specific job size distributions.

The bound obtained in Equation 8 is quite useful. In Section 6 we will use Equation 8 to prove Theorem 1 and then combine it with the HT property, to prove Theorem 2.

Below we use Equation 8 to prove Theorem 7.

THEOREM 7. *For any c.f.m.f.v. job size distribution and any load* $\rho < 1$,

$$E[S(x)]_{SRPT} \leq 2 + 2\rho$$

for all jobs of size x *such that* $\rho(x) \leq \frac{1}{2}$. *Hence, for job size distributions with the HT property, at least 99% of the jobs have an expected slowdown of at most 4, irrespective of the system load.*

PROOF. Follows directly from equation (8), Theorem 6. $\Box$

So far, we have shown two types of results with respect to starvation. We either show that *all* jobs do well for *most* loads. Or, *most* jobs do well for *all* loads. A natural question to ask at this point is, whether there are job size distributions for which all jobs do well at all loads.

We show that this is not the case. When load approaches 1, the largest job will perform worse under SRPT for any job size distribution (which has a well-defined largest job).

THEOREM 8. *For every c.f.m.f.v. bounded distribution,* $\exists \rho < 1$ *such that*

$$E[T(l)]_{SRPT} > E[T(l)]_{PS}$$

where l *is the size of the largest job.*

PROOF. We will lower bound $E[T(l)]_{SRPT}$ and show that $\exists\ \rho < 1$, such that $E[T(l)]_{SRPT} > \frac{l}{1-\rho} = E[T(l)]_{PS}$

The waiting time of the largest job under SRPT is simply $\frac{m_2(l)}{2(1-\rho)^2}$. To lower bound the residence time under SRPT we use the Chebyshev Integral Inequality, (12), with $y = x$, $u(x) = \frac{1}{1-\rho(x)}$, $v(x) = 1-\rho(x)$, $a = 0$ and $b = l$. Note that u and v satisfy the conditions in (12) since $\rho(x)$ is non-decreasing in x. Thus we get,

$$1 \leq \frac{\int_0^l (1-\rho(x))dx \int_0^l \frac{dx}{1-\rho(x)}}{l^2}$$

Equivalently,

$$\text{Residence Time(l)} = \int_0^l \frac{dx}{1-\rho(x)} \geq \frac{l^2}{l - l\rho + \lambda m_2(l)}$$

So,

$$\begin{aligned}
& E[T(l)]_{SRPT} - E[T(l)]_{PS} \\
\geq\ & \frac{\lambda m_2(l)}{2(1-\rho)^2} + \frac{l^2}{l-\rho l+\lambda m_2(l)} - \frac{l}{1-\rho} \\
=\ & \frac{\lambda m_2(l)}{1-\rho}\left(\frac{1}{2(1-\rho)} - \frac{1}{(1-\rho(1-d))}\right) \\
& [\text{ where } d = \frac{m_2(l)}{lE[X]}] \\
=\ & \frac{\lambda m_2(l)}{2(1-\rho)^2(1-\rho(1-d))}(\rho(1+d)-1)
\end{aligned}$$

Since both l and $E[X]$ are finite, $d > 0$. Thus for any $\rho < 1$, such that $\rho(1+d) > 1$, $E[T(l)]_{SRPT} - E[T(l)]_{PS} > 0$.

Hence, for any job size distribution, there is a load (less than 1) such that the largest job has a higher expected response time under SRPT than under PS. $\Box$

6. PROOF OF THEOREMS 1 AND 2

We will now use Theorem 6 to prove Theorem 1.

PROOF. (**Theorem 1**) Let $g(\rho(x))$ denote the improvement factor in Theorem 6.

$$g(\rho(x)) = \frac{1-\rho}{1-\rho(x)}\left[\frac{\rho}{2(1-\rho(x))}+1\right]$$

So, $E[T(x)]_{SRPT} \leq g(\rho(x))E[T(x)]_{PS}$.

$$\begin{aligned}
E[T]_{SRPT} &= \int_0^\infty E[T(x)]_{SRPT}\ f(x)dx \\
&\leq \int_0^\infty g(\rho(x))E[T(x)]_{PS}\ f(x)dx \\
&= \int_0^\rho \frac{1}{\lambda}\frac{g(\rho(x))}{1-\rho}d(\rho(x)) \\
& \quad [\text{Since}, d\rho(x) = \lambda x f(x)dx] \\
&= \frac{1}{\rho}E[T]_{PS} \int_0^\rho g(\rho(x))d(\rho(x))
\end{aligned}$$

Integrating g by parts, we get

$$\int_0^\rho g(\rho(x))d\rho(x) = \frac{\rho^2}{2} - (1-\rho)\log(1-\rho) \tag{10}$$

Thus, it follows that $E[T]_{SRPT} \leq h(\rho)E[T]_{PS}$.

For slowdown, we similarly obtain,

$$\begin{aligned} E[S]_{SRPT} &= \int_0^\infty E[S(x)]_{SRPT}\ f(x)dx \\ &\le \int_0^\infty E[S(x)]_{PS}\ g(\rho(x))f(x)dx \\ &\quad \text{[Dividing both sides of (8) by } x\text{]} \\ &= \int_0^\rho \frac{1}{1-\rho} g(\rho(x)) \frac{1}{\lambda x} d(\rho(x)) \qquad (11) \end{aligned}$$

Observe that $g(\rho(x))$ is increasing in $\rho(x)$ and $\frac{1}{x}$ is decreasing in $\rho(x)$. We now apply the Chebyshev Integral Inequality [16], which states that if $u(y), v(y)$ are non-negative functions which are non-decreasing and non-increasing respectively, then

$$(b-a)\int_a^b u(y)v(y)dy \le \int_a^b u(y)dy \int_a^b v(y)dy \qquad (12)$$

Setting $y = \rho(x), u(\rho(x)) = g(\rho(x)), v(\rho(x)) = \frac{1}{x}, a = 0$ and $b = \rho$ we get,

$$\begin{aligned} & E[S]_{SRPT} \\ \le\ & \frac{1}{1-\rho}\frac{1}{\rho}\int_0^\rho g(\rho(x))d\rho(x) \int_0^\rho \frac{d\rho(x)}{\lambda x} \\ =\ & \frac{1}{\rho}\int_0^\rho \frac{1}{1-\rho(x)}\left[\frac{\rho}{2(1-\rho(x))}+1\right]d\rho(x)\int_0^\infty f(x)dx \\ =\ & \frac{h(\rho)}{1-\rho} \end{aligned}$$

Thus, $E[S]_{SRPT} \le h(\rho)E[S]_{PS}$.

We now show that $h(\rho) \le 1$ for all $\rho \le 1$.

$$\frac{dh(\rho)}{d\rho} = \frac{1}{2} + \frac{1}{\rho} + \frac{\log(1-\rho)}{\rho^2}$$

Using the identity,

$$\log(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \dots$$

we get that

$$\frac{d(h(\rho))}{d\rho} \le 0, \quad \forall \quad 0 \le \rho \le 1$$

Thus $h(\rho)$ is decreasing in ρ. Observing that $h(0) = 1$, it follows that $h(\rho) \le 1$, for all ρ.

Thus $E[S]_{SRPT} \le E[S]_{PS}$. □

PROOF. (**Theorem 2**) For distributions with the HT property we know that it takes at least 99% of the smallest jobs to make up half the load. So, we split Equation 11 as

$$\begin{aligned} E[S]_{SRPT} &\le \int_0^{\frac{\rho}{2}} \frac{1}{1-\rho} g(\rho(x))\frac{1}{\lambda x}d(\rho(x)) \\ &+ \int_{\frac{\rho}{2}}^{\rho} \frac{1}{1-\rho} g(\rho(x))\frac{1}{\lambda x}d(\rho(x)) \end{aligned}$$

Now applying the Chebyshev Integral Inequality, we get

$$\begin{aligned} & E[S]_{SRPT} \\ \le\ & \frac{2}{\rho(1-\rho)}\left(\int_0^{\frac{\rho}{2}} g(\rho(x))d(\rho(x))\int_0^{\frac{\rho}{2}}\frac{1}{\lambda x}d(\rho(x))\right) \\ & + \frac{2}{\rho(1-\rho)}\left(\int_{\frac{\rho}{2}}^{\rho} g(\rho(x))d(\rho(x))\int_{\frac{\rho}{2}}^{\rho}\frac{1}{\lambda x}d(\rho(x))\right) \\ =\ & \frac{2}{\rho(1-\rho)}\left(\int_0^{\frac{\rho}{2}} g(\rho(x))d(\rho(x))\int_0^{x_h} f(x)dx\right) \\ & + \frac{2}{\rho(1-\rho)}\left(\int_{\frac{\rho}{2}}^{\rho} g(\rho(x))d(\rho(x))\int_{x_h}^{\infty} f(x)dx\right) \\ & \text{[where } x_h \text{ is such that } \rho(x_h) = \frac{\rho}{2}\text{]} \\ & \text{[Moreover, observe that } F(x_h) \ge 0.99\text{]} \\ \le\ & \frac{2}{\rho(1-\rho)}\left(\int_0^{\frac{\rho}{2}} g(\rho(x))d(\rho(x))\right) \\ & + \frac{2}{\rho(1-\rho)}\left(0.01\int_{\frac{\rho}{2}}^{\rho} g(\rho(x))d(\rho(x))\right) \\ \le\ & E[S]_{PS}\left(\frac{\rho(1-\rho)}{2-\rho} - \frac{2(1-\rho)}{\rho}\log(1-\frac{\rho}{2})\right) \\ & +0.01E[S]_{PS}\left(\frac{\rho}{2-\rho} - \frac{2(1-\rho)}{\rho}\log\frac{1-\rho}{1-\frac{\rho}{2}})\right) \\ =\ & E[S]_{PS}\left(\frac{\rho(1.01-\rho)}{2-\rho}\right) \\ & -E[S]_{PS}\left(\frac{2(1-\rho)}{\rho}(\log(1-\frac{\rho}{2}) + 0.01\log\frac{1-\rho}{1-\frac{\rho}{2}})\right) \end{aligned}$$

It is easy to see that the improvement factor approaches $\frac{1}{100}$ as $\rho \to 1$. □

7. OVERLOAD

We now look at the case when the system is overloaded ($\rho > 1$). That is, jobs arrive at a rate higher than the rate at which they can be worked upon. These situations often arise in real systems, and thus performance of scheduling policies under overload is an important factor in determining the goodness of a policy.

In the case of PS scheduling under overload, the mean response time for *every* job is infinite [14]. For SRPT scheduling under overload, to the best of our knowledge, no formula has been derived for the performance of a job of size x. We derive such a formula in this section, by combining ideas from [24] and [12].

THEOREM 9. *Given an M/G/1/SRPT system with load $\rho > 1$ and arrival rate λ. Assume a c.f.m.f.v.job size distribution with probability density function $f(t)$. Consider a job of size x such that $\rho(x) < 1$. Then*

$$\begin{aligned} E[T(x)]_{SRPT,\rho>1} &= \int_0^x \frac{dt}{1-\rho(t)} \\ &+ \frac{\lambda\int_0^x t^2 f(t)dt + \lambda(F(y)-F(x))x^2}{2(1-\rho(x))^2} \end{aligned}$$

where y is such that $\rho(y) = 1$.

Before we begin the proof, observe that the above formula is very similar to the original formula for $E[T(x)]_{SRPT}$ (not in overload) which we saw in Equation 1. The only difference is that the $1-F(x)$ term has been replaced with a $F(y)-F(x)$ term.

PROOF. The response time for a job of size x under SRPT can be split into waiting time and residence time, as shown in Section 4. Let us consider the derivation of the waiting time. It is shown in [24] that the waiting time for a job of size x in M/G/1/SRPT is equivalent to the waiting time for a job of size x in a particular non-preemptive priority system specified as follows: The non-preemptive priority system has 3 types of arrivals. Jobs of size $< x$ have priority 1 (highest priority). Jobs of size x have priority 2. Jobs of size $> x$ have priority 3. Jobs of priority 1 and 2 arrive according to the original Poisson Process. However jobs of priority 3 only arrive into the non-preemptive priority system at the moments when those jobs would have been reduced to size x in the original SRPT system. When the jobs of priority 3 arrive into the non-preemptive priority system, they arrive with size x. Since we are concerned with the waiting time for a type 2 job, the particular arrival process of jobs of priority 3 does not have any effect on the mean response time of a job of size x.

Now if $\rho < 1$, the formula for waiting time of a class c job in a non-preemptive system with k priority classes is given by

$$E[W_c] = \frac{\sum_{i=1}^{k} \lambda_i E[X_i^2]}{2(1-\sum_{i=1}^{c-1}\rho_i)(1-\sum_{i=1}^{c}\rho_i)} \quad (13)$$

where $E[W_c]$ is the expected waiting time for a job in class c, X_i is the job size distribution of the i^{th} class, λ_i is the arrival rate of jobs in the i^{th} class and the ρ_i is the load made up by jobs in the i^{th} class.

Observe that applying Equation 13 to the 3 class priority system described above gives the waiting time for SRPT as stated in Equation 2.

Phipps extended the result in Equation 13 for the case when $\rho > 1$, [12]. They show that for a job in class c such that $\sum_{i=1}^{c} \rho_i < 1$

$$E[W_c] = \frac{\sum_{i=1}^{k} v_i \lambda_i E[X_i^2]}{2(1-\sum_{i=1}^{c-1}\rho_i)(1-\sum_{i=1}^{c}\rho_i)} \quad (14)$$

where v_i is such that, $v_c = 1$ if $\sum_{i=1}^{c}\rho_i < 1$ and $v_c = 0$ if $\sum_{i=1}^{c-1}\rho_i > 1$. For the class c where $\sum_{i=1}^{c-1}\rho_i < 1$ and $\sum_{i=1}^{c}\rho_i \geq 1$, v_c is a number between 0 and 1 such that $\sum_{i=1}^{c} v_i\rho_i = 1$. Thus v_i indicates the fraction of jobs of each class which are executed in the steady state.

To obtain an expression for the waiting time under SRPT for $\rho > 1$, observe that the equivalence of the waiting time under SRPT and the three class non-preemptive system remains unchanged. Also observe that $\frac{F(y)-F(x)}{1-F(x)}$ fraction of jobs of size $> x$ are executed by SRPT in the steady state. Thus applying Equation 13 to the 3 class priority system with $\lambda_3 = (F_y - F_x)\lambda$, we obtain

$$E[W(x)]_{SRPT} = \frac{\frac{1}{2}\lambda\int_0^x t^2 f(t)dt + \frac{1}{2}\lambda(F(y)-F(x))x^2}{(1-\rho(x))^2}$$

where y is such that $\rho(y) = 1$, and $\rho(x) < 1$.

To obtain expression for the residence time we notice that once a job of size x begins execution and if its current size is t, then this job is affected only by load made up by jobs of size smaller than t. Thus the expression for the residence time under $\rho > 1$ remains the same as in Equation 3. This gives us the result for the response time under overload. □

To appreciate the above result, we consider the $B(k, p, \alpha = 1.1)$ job size distribution. We consider the mean time in system for a job of size x such that $\rho(x) = .5\rho$ and we let ρ range from 0 to 2. This job has exactly half the system load below it and half above. By the HT property, this job of size x is in the 99th %-tile of the job size distribution. The expected slowdown for x is shown in Figure 2 in the case of PS scheduling and in the case of SRPT scheduling. Observe that in the case of PS scheduling, x has infinite mean slowdown once the system load reaches 1. However, under SRPT scheduling, x has finite mean slowdown up until the point where $\rho(x) = 1$, i.e. up until $\rho = 2$.

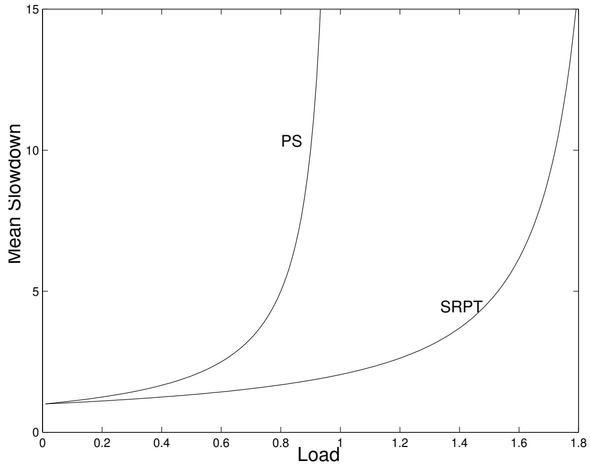

Figure 2: Mean slowdown for a job in the 99 percentile of the $B(k, p, \alpha = 1.1)$ job size distribution, as a function of ρ.

In particular, Figure 2 shows that when the system load is 1.5 (respectively 1.8), the mean slowdown of a job in the 99%-tile of the job size distribution is only 4 (respectively 15) for SRPT scheduling as compared with infinity for PS scheduling.

The above results help explain our trace-based experimental results. Under overload conditions, PS simply stalls: all jobs experience infinite mean slowdown. However under the same overload conditions, SRPT keeps getting jobs out.

8. PREEMPTION OVERHEAD

When implementing a scheduling policy with preemptions (like SRPT, PS, FB ...) the overhead associated with preemptions is a cause for concern. In real systems, PS is implemented as round robin where each process gets a finite time quantum. The number of preemptions depends on the size of this time quantum.

However, under SRPT, the amortized number of preemptions per job is at most two. This is true irrespective of the arrival process. To see this, observe that a job may be preempted under SRPT only when a new job arrives into the system or an existing job is completed. Since any job arrives and completes exactly once, then total number of preemptions is never more than twice the number of job arrivals.

By comparison, under any reasonable implementation of PS the average number of preemptions per job will be more than two, since most jobs requires more than two time quantums of service.

9. COMPARISON OF SRPT TO OTHER SCHEDULING POLICIES

Throughout this paper we compared SRPT scheduling with PS scheduling only and ignored how SRPT might compare to other policies. In this section we will consider some other commonly used policies and explain why we dismissed these. In particular we consider, First Come First Serve (FCFS), Last Come First Serve (LCFS), Random, Non-preemptive Shortest Job First (SJF), Foreground-Background (FB) and Preemptive Last Come First Serve (P-LCFS).

We will argue that non-preemptive policies have far worse mean performance than SRPT, under more variable job size distributions. This is the reason why we've ignored such policies in this paper, although, as we point out, these policies do have some nice properties for a few of the very largest jobs.

We will then argue that each of the preemptive policies above can easily be shown to either be no better than PS on all jobs, or no better than SRPT on all jobs. This explains why we've ignored these policies in the paper as well.

Throughout we assume an M/G/1 queue.

9.1 Non-preemptive policies

It is well known [3] that the expected response time for a job of size x is the same for *all* non-preemptive policies which do not make use of size. This includes for example FCFS, LCFS and Random. This time is given by

$$E[T(x)]_{FCFS,LCFS,RANDOM} = \frac{\lambda \int_0^\infty t^2 f(t)dt}{2(1-\rho)} + x \quad (15)$$

A different kind of non-preemptive policy which makes use of size and favors smaller jobs as compared with longer ones is SJF. For a c.f.m.f.v. job size distribution f, the expected response time for a job of size x under SJF is given by

$$E[T(x)]_{SJF} = \frac{\lambda \int_0^\infty t^2 f(t)dt}{2(1-\rho(x))^2} + x \quad (16)$$

Observe that in both Equations 15 and 16, the queueing time for a job of size x is dependent on the second moment of the entire job size distribution. This is especially punitive for small jobs. By contrast, observe that under SRPT (see Equation 1) the time in system for a job of size x depends only up to the second moment of the distribution truncated at x.

Thus in the case of a highly variable distribution, we would expect that the mean response time will be significantly higher for these non-preemptive policies as compared with that under SRPT. To illustrate this point, Figure 3 shows the mean response time for these nonpreemptive policies as compared with SRPT when the job size distribution is $B(k, p, \alpha = 1.1)$. Observe that SRPT improves upon these other policies by several orders of magnitude.

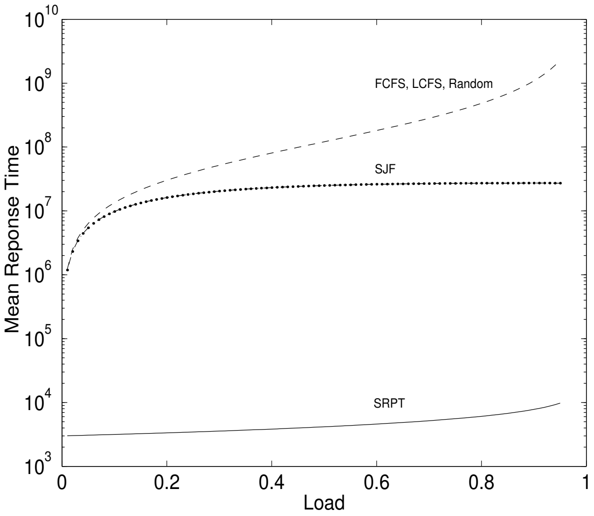

Figure 3: ***The figure shows the Mean Response Time as a function of load under FCFS, LCFS, Random, SJF and SRPT for the Bounded Pareto job size distribution,*** $B(k, p, \alpha = 1.1)$.

While non-preemptive policies have very high mean response time as compared with SRPT, these non-preemptive policies might actually improve upon SRPT with respect to response time of just the very large jobs. The point is that, given a sufficiently large job, one could imagine its waiting time (time until first idle period) becoming negligible compared with its size. Thus the response time would be dominated by the residence time. The residence time will be 1 under a non-preemptive policy, but could approach $\frac{1}{1-\rho}$ for very large jobs under SRPT.

In the above scenario, however, it is typically only the truly largest jobs which benefit under a non-preemptive policy, as compared with SRPT. For example, under the $B(k, p, \alpha = 1.1)$ job size distribution, at a load of 0.9, only about 0.00005% of the jobs have a lower expected response time under FCFS as compared with SRPT and only 0.0005% of the jobs have a lower response time under SJF as compared with SRPT.

9.2 Preemptive policies

In addition to PS, the other common preemptive policies are Foreground-Background (FB) and Preemptive Last Come First Serve (P-LCFS).

For a c.f.m.f.v. job size distribution f, the expected response time for a job of size x under FB and P-LCFS is respectively given by

$$E[T(x)]_{FB} = \frac{\lambda(m_2(x) + x^2(1 - F(x))}{2(1 - \rho(x))^2} + \frac{x}{1 - \rho(x)} \quad (17)$$

$$E[T(x)]_{P-LCFS} = \frac{x}{1 - \rho} \quad (18)$$

Observe that under P-LCFS the expected response time for a job of size x is the same as that under PS. Thus it suffices to compare SRPT with PS.

The comparison of FB scheduling with SRPT scheduling is slightly more interesting. Comparing equations 17 and 1 we observe that the first term of Equation 17 is identical to the expected waiting time for every job under SRPT. Secondly the expression for the residence time under SRPT can easily be seen to be less than or equal to the second term in Equation 17. Thus, we have the following observation.

OBSERVATION 1. *For any job size distribution and for any load $\rho < 1$, every job has a higher expected response time (hence slowdown) under FB scheduling as compared with SRPT scheduling.*

It follows that both the mean slowdown and the mean response time will be worse under FB as compared with SRPT. Also, the unfairness to large jobs will be higher under FB than under SRPT. Of course, FB has the advantage over SRPT that it does not require knowledge of job size.

10. CONCLUSION

The goal of this paper is to dismiss notions of "unfairness" commonly associated with SRPT. We prove that under moderate system load, for *any* job size distribution, *all* jobs prefer SRPT to PS. As the load increases, this statement is only true for job size distributions with the heavy-tailed property. However the situation is not as bad as one might think for general distributions. Even under conditions of higher load, for general distributions, we show that the majority of jobs are insensitive to the higher load. For the remaining jobs, we prove absolute bounds on how high the expected slowdown under SRPT can be as compared with PS.

While it is well-known that SRPT is optimal with respect to mean response time, the degree of the improvement in mean response time and mean slowdown of SRPT over PS has not been studied for general job size distributions. We obtain bounds on the mean improvement factor of SRPT over PS under general job size distributions and under job size distributions with the heavy-tailed property.

We also obtain closed-form expressions for the performance of SRPT under overload. We show that under overload the difference between SRPT and PS is even more dramatic.

Acknowledgements

We would like to thank Mark Squillante and the anonymous Sigmetrics reviewers for their valuable comments.

11. REFERENCES

[1] M. Bender, S. Chakrabarti, and S. Muthukrishnan. Flow and stretch metrics for scheduling continous job streams. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[2] L. Cherkasova. Scheduling strategies to improve response time for web applications. In *High-performance computing and networking: international conference and exhibition*, pages 305–314, 1998.

[3] Richard. W. Conway, W. L. Maxwell, and Louis W. Miller. *Theory of Scheduling*. Addison-Wesley Publishing Company, 1967.

[4] M. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. In *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 160–169, May 1996.

[5] J.E. Gehrke, S. Muthukrishnan, R. Rajaraman, and A. Shaheen. Scheduling to minimize average stretch online. In *40th Annual symposium on Foundation of Computer Science*, pages 433–422, 1999.

[6] Mor Harchol-Balter, Nikhil Bansal, and Bianca Schroeder. Implementation of SRPT scheduling in web servers. Technical Report CMU-CS-00-170, Carnegie Mellon School of Computer Science, October 2000.

[7] Mor Harchol-Balter, M. Crovella, and S. Park. The case for srpt schduling in web servers. Technical Report MIT-LCS-TR-767, MIT Lab for Computer Science, October 1998.

[8] Mor Harchol-Balter, Mark Crovella, and Cristina Murta. On choosing a task assignment policy for a distributed server system. *IEEE Journal of Parallel and Distributed Computing*, 59:204 – 228, 1999.

[9] Mor Harchol-Balter and Allen Downey. Exploiting process lifetime distributions for dynamic load balancing. In *Proceedings of SIGMETRICS '96*, pages 13–24, 1996.

[10] G. Irlam. Unix file size survey - 1993. Available at `http://www.base.com/gordoni/ufs93.html`, September 1994.

[11] A. Jean-Marie and P. Robert. On the transient behavior of the processor-sharing queue. *Queueing Systems: Theory and Applications*, 17:129–136, 1994.

[12] T.E. Phipps Jr. Machine reapir as a waiting line problem. *Operations Research*, 4:76–86, 1956.

[13] L. Kleinrock, R.R. Muntz, and J. Hsu. Tight bounds on average response time for time-shared computer systems. In *Proceedings of the IFIP Congress*, volume 1, pages 124–133, 1971.

[14] Leonard Kleinrock. *Queueing Systems*, volume II. Computer Applications. John Wiley & Sons, 1976.

[15] W. E. Leland and T. J. Ott. Load-balancing heuristics and process behavior. In *Proceedings of Performance and ACM Sigmetrics*, pages 54–69, 1986.

[16] D.S. Mitrinovic. *Anayltic Inequalities*. Springer-Verlag, 1970.

[17] E. Modiano. Scheduling algorithms for message transmission over a satellite broadcast system. In *Proceedings of IEEE MILCOM '97*, pages 628–634, 1997.

[18] A.V. Pechinkin, A.D. Solovyev, and S.F. Yashkov. A system with servicing discipline whereby the order of remaining length is serviced first. *Tekhnicheskaya Kibernetika*, 17:51–59, 1979.

[19] R. Perera. The variance of delay time in queueing system M/G/1 with optimal strategy SRPT. *Archiv fur Elektronik und Uebertragungstechnik*, 47:110–114, 1993.

[20] David L. Peterson and David B. Adams. Fractal patterns in DASD I/O traffic. In *CMG Proceedings*, December 1996.

[21] J. Roberts and L. Massoulie. Bandwidth sharing and admission control for elastic traffic. In *ITC Specialist Seminar*, 1998.

[22] R. Schassberger. The steady-state appearance of the M/G/1 queue under the discipline of shortest remaining processing time. *Advances in Applied Probability*, 22:456–479, 1990.

[23] L.E. Schrage. A proof of the optimality of the shortest processing remaining time discipline. *Operations Research*, 16:678–690, 1968.

[24] L.E. Schrage and L.W. Miller. The queue M/G/1 with the shortest processing remaining time discipline. *Operations Research*, 14:670–684, 1966.

[25] F. Schreiber. Properties and applications of the optimal queueing strategy SRPT - a survey. *Archiv fur Elektronik und Uebertragungstechnik*, 47:372–378, 1993.

[26] A. Silberschatz and P. Galvin. *Operating System Concepts, 5th Edition.* John Wiley & Sons, 1998.

[27] D.R. Smith. A new proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 26:197–199, 1976.

[28] W. Stallings. *Operating Systems, 2nd Edition.* Prentice Hall, 1995.

[29] A.S. Tanenbaum. *Modern Operating Systems.* Prentice Hall, 1992.

[30] Ronald W. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice Hall, 1989.