Tail asymptotics for policies favoring short jobs in a many-flows regime

Chang-Woo Yang
University of Texas at Austin
Austin, TX 78712
cyang@ece.utexas.edu

Adam Wierman
Carnegie Mellon University
Pittsburgh, PA 15213
acw@cs.cmu.edu

Sanjay Shakkottai
University of Texas at Austin
Austin, TX 78712
shakkott@ece.utexas.edu

Mor Harchol-Balter
Carnegie Mellon University
Pittsburgh, PA 15213
harchol@cs.cmu.edu

ABSTRACT

Scheduling policies that prioritize short jobs have received growing attention in recent years. The class of SMART policies includes many such disciplines, e.g. Shortest-Remaining-Processing-Time (SRPT) and Preemptive-Shortest-Job-First (PSJF). In this work, we study the delay distribution of SMART policies and contrast this distribution with that of the Least-Attained-Service (LAS) policy, which indirectly favors short jobs by prioritizing jobs with the least attained service (age).

We study the delay distribution (rate function) of LAS and the SMART class in a discrete-time queueing system under the many sources regime. Our analysis in this regime (large capacity and large number of flows) hinges on a novel two dimensional queue representation, which creates tie-break rules. These additional rules do not alter the policies, but greatly simplify their analysis. We demonstrate that the queue evolution of all the above policies can be described under this single two dimensional framework.

We prove that all SMART policies have the same delay distribution as SRPT and illustrate the improvements SMART policies make over First-Come-First-Served (FCFS). Furthermore, we show that the delay distribution of SMART policies stochastically improves upon the delay distribution of LAS. However, the delay distribution under LAS is not too bad – the distribution of delay under LAS for most jobs sizes still provides improvement over FCFS. Our results are complementary to prior work that studies delay-tail behavior in the large buffer regime under a single flow.

1. INTRODUCTION

Recently there has been an increased focus on the scheduling policies (disciplines) used in computer systems. With the goal of improving user perceived delay, policies that bias towards jobs with small sizes (service demands), e.g. SRPT and PSJF, are increasingly being presented as alternatives to standard policies, e.g. Processor-Sharing (PS) and FCFS. For example, variations of SRPT have been shown to provide dramatic improvements for mean delay in web servers [9, 20], and other size based based policies have been applied in an array of applications such as databases, supercomputing centers, and disks. Even in applications where the job sizes are unknown, system designers have suggested policies such as LAS[1], which prioritizes jobs with small ages (attained service) so that small jobs (which always have small ages) tend to have the server to themselves. For example, in routers, variations of LAS have been shown to dramatically reduce delay [18, 19].

Given the increased focus on the application of size based policies by practitioners, there has also been a growing amount of analytic research studying policies that bias towards small jobs. The focus of this work has been primarily on SRPT and LAS due to the fact that SRPT is known to be optimal for mean delay [23], and LAS is known to be optimal for mean delay among policies that are blind to job sizes when the service distribution has a decreasing failure rate [21, 22]. However, SRPT and LAS are idealized versions of the policies implemented in computer systems. Implementation constraints force the use of more complex hybrid policies in practice [20, 19, 14], though these hybrid policies still obey the general heuristic of biasing towards small jobs. In order to formalize this heuristic, the class of SMART policies was recently introduced [26]. The SMART class includes SRPT and PSJF in addition to other hybrid policies (excluding LAS), while still guaranteeing that the mean delay of any SMART policy is near optimal under all service distributions. For a formal definition of SMART see Section 5.

Recent studies of policies that prioritize small job sizes, such as LAS, SRPT, and the SMART class, have focused on the delay experienced by a job of size k, $W(k)$. The interest in $W(k)$ is spurred by the a desire to understand how the prioritization of small job sizes affects the behavior of $W(k)$ across jobs sizes, k. In particular, there are worries about the delay experienced by large k, which are biased against. Much attention has been given to understanding $E[W(k)]$, the mean delay experienced by a job of size k, across k [9, 18, 19, 25]; however far less is understood about the *distribution* of $W(k)$ under LAS and the SMART class.

The difficulty in direct analysis of the distribution of $W(k)$ has led researchers to study asymptotic scalings of the distribution. The most common scaling is referred to as the *large buffer* large deviations framework, which studies the tail behavior of $W(k)$, i.e.

SIGMETRICS'06, June 26–30, 2006, Saint-Malo, France.
Copyright 2006 ACM ...$5.00.

[1]LAS is known in the literature under a variety of other names including: Foreground-Background (FB) and Shortest-Elapsed-Time first (SET).

$Pr(W(k) > m)$, for large m in the GI/GI/1 queue [2, 6, 17]. In this framework, the tail of $W(k)$ under LAS and the SMART class have been shown to behave proportionally to a busy period where the job size distribution is truncated at k. Under SMART policies, jobs of size larger than k do not contribute to the busy period; whereas under LAS jobs of size larger than k contribute k to the busy period [12, 16, 15]. This is in contrast to the behavior of FCFS, where the tail of $W(k)$ is proportional to the tail of the stationary workload for all k. While the large buffer framework has provided many insights about the behavior of the tail of $W(k)$, it provides little information about $Pr(W(k) > m)$ for small m.

In this paper, we consider a different asymptotic scaling of the distribution of $W(k)$. Motivated by applications such as high traffic web servers and routers that have enormous available bandwidth and thousands of simultaneous flows, we consider the *many sources* large deviations framework. The many sources framework scales the number of arrival flows, the buffer size, and the service capacity proportionally, as shown in Figure 1.

We have three main motivations for studying the many sources framework. First, the many sources framework considers a very large number of flows and, as a result, captures effects such as statistical multiplexing that the large buffer framework does not. Second, the characterization of the critical event that leads to delay under many sources framework is complementary to that of the large buffer framework. In the large buffer framework the critical event of a large delay is typically the arrival of a handful of very large jobs from a single flow, while in the many sources framework the critical event characterizes a large burst of arrivals across many flows. Thus, the critical event in the many sources regime can be useful in settings where the critical event in the large buffer regime is inappropriate. For example, a high traffic web server or router with enormous bandwidth that is accessed by a large number of flows cannot be swamped by a handful of large arrivals from any single flow, rather the critical event is a burst of arrivals from a large number of flows. The third motivation for considering the many sources framework is that, under the many sources scaling, it is possible to characterize the critical event that leads to $\Pr(W(k) > m)$ for any finite delay m, as opposed to only the case of $m \to \infty$ as in the large buffer framework.

Summary of Contributions

We prove three main results under the many sources framework in this paper. First, we prove that the decay rate of the delay experienced by size k jobs is the same under all policies in the SMART class (Theorem 1). Thus, we show that all SMART policies are equivalent in the many sources framework. Second, we derive the decay rate of the delay for a job of size k under LAS (Theorem 2). Third, we prove that all SMART policies stochastically outperform LAS with respect to delay for any job size (Corollary 3). Prior to these results, very few scheduling policies have been analyzed in the many sources framework. To the best of our knowledge, only the decay rates of FCFS [3], a simple priority queueing system [8, 24], Generalized Processor Sharing [10], and SRPT [27] have been reported in literature.

Our results are enabled through the use of a new analytic framework that we refer to as the *two dimensional queueing framework*. This framework adds tie-break rules to policies in a way that does not alter the asymptotic performance of the policies, but greatly simplify their analysis (see Section 4). The strength of this novel framework is that it enables: *(i)* the study of policies that depend on the job state (age and/or remaining size), as opposed to only the queue length; and *(ii)* the study of a *class of policies*, as opposed to only the analysis of individual policies. Using the two dimensional queueing framework, we are able to conclude that the effect of preemption becomes negligible under SMART policies in the many sources regime, thus allowing us to treat the class as a more tractable priority queuing discipline. For LAS, the two dimensional queueing framework adds a secondary ordering scheme which allows a more detailed snapshot of the queue state at any time, and thus makes the system analysis tractable.

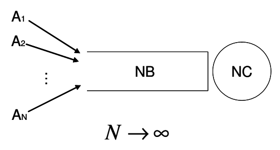


Figure 1: The many sources large deviation framework.

In order to explore the behavior of the decay rates that we derived for the SMART class and LAS, we perform simulations and numeric calculations in Section 7. Using simulations we illustrate that the distribution of $W(k)$ converges to the many sources asymptote very quickly: convergence is achieved after only 20 flows. Considering that high traffic web servers and routers routinely have many more than 20 simultaneous flows, this provides practical motivation for the many sources scaling. Further, we use numeric calculations to compare the decay rate of $W(k)$ under SMART and LAS across large and small job sizes, and to study the penalty LAS pays (compared with SMART policies) for not using job sizes to prioritize.

2. PRELIMINARIES

In this section we will introduce the many sources framework, which we will use throughout the paper. We consider a queueing system with a single queue and a single server having stationary and ergodic arrival and service processes, where the arrival and service processes are independent of each other. The system operates in discrete time, i.e. a batch of jobs arrive at the beginning of each time slot and jobs are serviced at the end of each time slot. The queue state is measured immediately after the service – just before the arrivals of the next time slot. We further assume that the possible sizes of the jobs are restricted to bounded multiples of a unit size. Thus, we represent the set of possible job sizes as $\mathcal{M} = \{1,2,3,\ldots,M\}$. The assumption that the service distribution is bounded is natural given the numerous studies that have observed that file sizes at web servers typically follow a bounded, highly variable distribution size[1, 4, 5].

In the many sources framework, the number of arrival processes is scaled along with the capacity of the system and the buffer size as depicted in Figure 1. Formally, for each job size $k \in \mathcal{M}$, we assume N independent, identically distributed processes. Further, we assume independence between arrival processes of different sized jobs. Notice that job arrivals from a single stream of any given size can be correlated across time-slots.

Equivalently, this model can be represented as a job arrival process that can be decomposed into independent arrival processes according to the job size. We define $A^N(a,b)$ as the total number of arrivals by all N arrival processes in the time-interval (a,b), where $a \leq b$[2]. For example, $A^N(0,0)$ signifies the total number of arrivals in time slot 0. We define $A_k^N(a,b)$ as the total number of jobs of size k that arrive in the queue during time-interval (a,b). Thus, the volume of size k arrivals arrivals is $kA_k^N(a,b)$. Further, we have $A^N(a,b) = \sum_{k=1}^{M} A_k^N(a,b)$. We assume that the capacity of the server, C, is scaled in proportion to the load, and at

[2]The notation (a,b) refers to time slots $\{a, a+1, \ldots, b\}$.

most NC unit data can be service at any time slot. We assume the stability of the queue, i.e.

$$E\left[\sum_{i=1}^{M} iA^N(a,b)\right] < NC(b-a-1) \quad \forall a < b.$$

Our goal is to study the tail probability of conditional delay in the many sources regime. The conditional *delay*, $W^{(N)}(k)$, is the delay experienced by the last job with size k in an arrival burst to a stationary system. The tail probability of delay, $\Pr(W^{(N)}(k) > m)$, is the probability that the last job of size k in the burst arriving at time slot l does not leave the system by the end of time slot $l+m$. It has been shown that, in the large deviation framework, the tail probability of delay decays as

$$\Pr(W^{(N)}(k) \geq m) = g(k,m)^N e^{-NI_W(k,m)},$$

under general conditions, where $\lim_{N\to\infty} -\frac{1}{N}\log g(k,m)^N = 0$. In other words, the most dominant trend of the tail probability is the exponential decay $I_W(k,m)$, which is appropriately called the *decay rate*. The decay rate of delay is defined as follows.

$$I_W(k,m) = \lim_{N\to\infty} -\frac{1}{N}\log\Pr(W^{(N)}(k) > m) \quad (1)$$

Note that the delay distribution for a job of size k depends on the capacity C, the threshold value m, the job size k, and the arrival process, $A_k^N(a,b)$, i.e. the decay rates of the arrival process, which has been well analyzed:

$$I_{A_k}^{(a,b)}(x) = \lim_{N\to\infty} -\frac{1}{N}\log\Pr(A_k^N(a,b) > Nx) \quad (2)$$

Alternatively, (2) can be derived using the moment generating function of the random variable $A_k^N(a,b)$ as follows.

$$I_{A_k}^{(a,b)}(x) = \sup_{\theta}\left(\theta x - \Lambda_{A_k}^{(a,b)}(\theta)\right) \quad (3)$$

where $\Lambda_{A_k}^{(a,b)}(\theta) = \lim_{N\to\infty}\frac{1}{N}\log E\left(e^{\theta A_k^N(a,b)}\right)$.

At this point, it is important to digress and observe the difference between (1) and the corresponding definition of the decay rate of conditional delay,$\gamma(W(k))$, in the large buffer large deviations framework:

$$\gamma(W(k)) = \lim_{m\to\infty} -\frac{1}{m}\log\Pr(W(k) > m) \quad (4)$$

Observe that the large buffer framework only considers a single arrival flow and derives the decay rate for $m\to\infty$, while the many sources framework calculates the decay rate under a large number of flows assumption ($N\to\infty$) for all $m>0$.

Though our goal is to study the decay rate of delay, the direct derivation is difficult, so we first consider the distribution of the *virtual delay*. The *virtual delay*, $V^{(N)}(k)$, is the delay seen by a fictitious (virtual) job that arrives at Q_k, queue for size k jobs, at the end of an arrival burst at $t=0$ (given that the system started at $t=-\infty$). The event $\left\{V^{(N)}(k) > m\right\}$ corresponds to a fictitious job arriving at the end of an arrival burst during time slot 0 and not departing the system until the mth time slot. Note that this setup ensures that the system is stationary at the arrival of the virtual job. To avoid confusion, we will refer to the delay as the *actual delay* in order to distinguish it from the *virtual delay*. Observe that the virtual delay is different from actual delay: for example, even when there is no arrival the virtual delay can be measured, whereas the actual delay is not defined. The decay rate of the virtual delay is defined as:

$$I_V(k,m) = \lim_{N\to\infty} -\frac{1}{N}\log\Pr(V^{(N)}(k) > m) \quad (5)$$

In our proofs, it will be necessary to have a more general definition of the decay rate than we have defined so far. Thus, for any sequence of rare events $\mathbf{H}^N$, we define

$$\mathbf{I}(\mathbf{H}) = \lim_{N\to\infty} -\frac{1}{N}\log\Pr\left(\mathbf{H}^N\right),$$

as the decay rate of a general sequence of events $\mathbf{H}^N$ whose probability becomes increasingly small as the system scales.

3. RESULTS AND DISCUSSION

The main results in this paper are the derivations of the decay rates for (i) the class of SMART policies and for (ii) LAS. In this section we will present and briefly discuss these results. The derivations of the results are postponed to later sections.

3.1 SMART

Our first theorem describes the decay rate of the SMART class. The SMART class, introduced in [26], is a class of disciplines that all bias towards jobs that were originally small and/or have small remaining service requirement. We defer the formal definition of SMART to Section 5, and for now just state that SMART contains many common policies, e.g. SRPT and PSJF, in addition to a wide array of hybrid policies with more complicated prioritization schemes. One important motivation for working with the class of SMART policies is to illustrate the wide range of policies that behave like SRPT. We will show that, under the many sources framework, all SMART policies behave equivalently; thus proving a strong link between the behavior of SRPT and the SMART class.

Our analysis of SMART hinges on coupling the queue dynamics of a SMART policy to the queue dynamics of a two-level priority queueing system. A two-level priority queueing system consists of a pair of queues (high/low priority), where jobs in the low priority queue are served in a FCFS manner only if the high priority queue is empty. Using the two dimensional queueing framework, we show the effect of any SMART policy is to *partition* the two dimensional collection of queues into three groups in any given time slot: a set with higher priority, a set with lower priority, and a set with unknown priority. This partition enables us to adapt the analysis for priority queues in order to analyze all SMART policies. For this purpose, we define a preemptive priority queueing system (PRI) where there are M queues, with jobs of original size k arriving to queue-k. Queues corresponding to smaller packets are granted higher priority over larger packets. Further, jobs can not switch queues and each queue is served using FCFS.

THEOREM 1. *Let $\epsilon > 0$. For any $k \in \mathcal{M}$, the decay rate of delay for a size k job under any SMART policy, $I_{\overline{W}}(k,m)$, satisfies*

$$I_{V_{C-\epsilon}}(k,m) \leq I_{\overline{W}}(k,m) \leq I_{V_C}(k,m), \quad (6)$$

where $I_{V_\mu}(k,m)$ is the virtual decay rate of delay under a priority queueing system, PRI, with capacity $N\mu$ and

$$I_{V_\mu}(k,m) = \inf_{T\geq 0}\left[\inf_{\vec{z}:\mathcal{Z}}\left\{\mathfrak{A}_{<k}(\vec{z}) + \mathfrak{A}_k\right\}\right], \quad (7)$$

where condition $\mathcal{Z}$ states that $\sum_{i=1}^{k} iz_i = \mu(T+m+1)$. Further,

$$\mathfrak{A}_{<k}(\vec{z}) = \sum_{i=1}^{k-1} I_{A_i}^{(-T,m)}(z_i)$$
$$\mathfrak{A}_k = I_{A_k}^{(-T,0)}(z_k)$$

This theorem states that asymptotically (in the large capacity and large number of flows regime), *all SMART policies behave alike*, in that their delay decay rates are the same. In other words, for a job of original size k and for any fixed integer $m \geq 0$, we have that the delay distribution for $\overline{W}^{(N)}(k)$ is given by

$$P\left(\overline{W}^{(N)}(k) > m\right) = g(k,m)^N e^{-NI_{\overline{W}}(k,m)},$$

where $I_{\overline{W}}(k,m)$ *is the same for all SMART policies.* Thus, the decay rate of any SMART policy is the same as that of SRPT, which was derived in [27].

The decay rate in (7) appears complicated, but does have intuition. It can be shown that in the many sources asymptote, the decay rate depends on the "most likely" way that the arrival processes deviate from their mean arrival rates in order to cause delay exceeding m. Thus, the two infimums choose the most likely time scale (T) and partition of the overall arrival rate to job sizes ($\vec{z}$) respectively. Then, inside the infimums, $\mathfrak{A}_{<k}(\vec{z})$ and $\mathfrak{A}_k$ characterize the delay caused to a size k job by jobs arriving over the time interval $(-T, m)$ with size $< k$ and by jobs arriving over the time interval $(-T, 0)$ with size k.

To illustrate this intuition, we now consider the special case when jobs are one of two sizes (1 or M). For this special case, we can provide simplified expressions which relate the delay distributions of jobs to the arrival process statistics. As a baseline for comparison, we compare with the delay distribution of FCFS [3].

For FCFS, it follows from the results in [3] that

$$\Pr(W^{(N)}(k) > m) \propto \max_{T\geq 0} \Pr(D^{(N)}(T) > C(T+m+1)) \tag{8}$$

for $k = 1$ and $k = M$ and where $a \propto b$ means that a and b have the same decay rate. Here $D^{(N)}(T)$ is the cumulative workload (including both size 1 and size M jobs) that arrives to the server over the time-interval $(-T, 0)$, i.e.,

$$D^{(N)}(T) = A_1^N(-T,0) + MA_M^N(-T,0)$$

Recall that the server capacity is C units per time-slot. The above expression (8) states that the probability that a job of size 1 or M experiences a delay of at least m is the same as the probability that *the cumulative arrival workload over a time interval of* $T+1$ *slots exceeds the cumulative server capacity over a time interval of* $T+m+1$ *slots,* for some fixed value of T. The maximizing value of T in the RHS of (8) is sometimes referred to as the critical time scale of the queue. Note that the decay rates for size 1 and M jobs are the same, since the only difference is their service requirement which is negligible in the large deviation framework.

Moving to SMART, it follows from Theorem 1 that

$$\Pr(W^{(N)}(k) > m) \propto \max_{T\geq 0} \Pr(E_k^{(N)}(T) > C(T+m+1)) \tag{9}$$

where

$$E_k^{(N)}(T) \begin{cases} A_1^N(-T,0), & k=1; \\ A_1^N(-T,m) + MA_M^N(-T,0), & k=M. \end{cases}$$

Note that for size 1 jobs, the above is the "best possible" delay distribution that can be achieved over the class of all work conserving policies. However, in the case of size k jobs for each fixed T, $E_M^{(N)}(T) \geq D^{(N)}(T)$, which immediately implies that the delay of a job of original size M with SMART stochastically dominates (i.e., is larger in a distributional-sense) the corresponding delay with FCFS. However, for heavy-tailed arrivals, it can be shown that this difference is small, of order $O(1/M)$, by observing that the decay rate of SMART matches that of SRPT, which has been compared to FCFS in [27]. This means that in the large N and M regime (i.e. large number of flows, and a large difference in arriving job sizes), the delay experienced by a size M job is similar under FCFS and SMART, while size 1 jobs experience far less delay under SMART than under FCFS.

3.2 Least Attained Service

LAS is a preemptive scheduling policy that shares the server evenly among the jobs in the system with the least attained service (age). Note that a newly arriving job always preempts the job (or jobs) currently in service and retains the processor until one of the following occurs: (i) the job departs, (ii) the next arrival appears, or (iii) the job has obtained an amount of service equal to that received by the job(s) preempted on arrival.

LAS is an important policy because it provides an approximation of SRPT in the case when job sizes (processing times) are not known, which is especially relevant for applications such as routers and operating systems where variations of LAS have been used in practice [18, 19]. LAS can be viewed as an approximation of SRPT because (i) small jobs need not queue behind large jobs and (ii) in many cases the age of a job is a good indicator of its remaining size. However, without use of job sizes, LAS cannot bias as strongly towards small job sizes as SMART policies. Thus, our goal in studying the decay rate of LAS is to understand the penalty LAS pays for not using job sizes to prioritize.

THEOREM 2. *Under Assumptions 1 and 2 (some reasonable technical assumptions to be defined in Section 6.2), the decay rate of delay for size k jobs under LAS,* $I_{\hat{W}}(k,m)$*, is*

$$I_{\hat{W}}(k,m) = \inf_{T\geq 0}\left[\inf_{\vec{y}:\mathcal{Y}} (\mathfrak{A}_{<k}(\vec{y}) + \mathfrak{A}_k(\vec{y}) + \mathfrak{A}_{>k}(\vec{y}))\right], \tag{10}$$

where condition $\mathcal{Y}$ *states that* $\sum_{i\in\hat{\mathbf{k}}} iy_i + ky_k + \sum_{i\in\check{\mathbf{k}}}(k-1)y_i = C(T+m+1)$ *with* $\hat{\mathbf{k}}\{1,\ldots,k-1\}$, $\check{\mathbf{k}} = \{k+1,\ldots,M\}$*, and* $y_k^{(1)} + \frac{k-1}{k}y_k^{(2)} = y_k$*. Further,*

$$\begin{aligned} \mathfrak{A}_{<k}(\vec{y}) &= \sum_{i=1}^{k-1} I_{A_i}^{(-T,m)}(y_i) \\ \mathfrak{A}_k(\vec{x}) &= I_{A_k}^{(-T,0)}(y_k^{(1)}) + I_{A_k}^{(1,m)}(y_k^{(2)}) \\ \mathfrak{A}_{>k}(\vec{y}) &= \sum_{j=k+1}^{M} I_{A_i}^{(-T,m)}(y_j) \end{aligned}$$

Theorem 2 characterizes the delay distribution of LAS in the many sources regime. Though the form of (10) is complicated, we can obtain intuition for it. Again, the decay rate depends on the "most likely" way that the arrival processes deviate from their mean arrival rates in order to cause delay exceeding m. Thus, the two infimums choose the most likely time scale (T) and arrival rates for each job size ($\vec{y}$), where y_k is separated into the arrivals before ($y_k^{(1)}$) and after ($y_k^{(2)}$) the tagged arrival. Then, inside the infimums, $\mathfrak{A}_{<k}(\vec{y})$, $\mathfrak{A}_k(\vec{y})$, and $\mathfrak{A}_{>k}(\vec{y})$ characterize the contribution to the delay of a size k job made by jobs with size $< k$, other jobs of size k, and jobs of size $> k$ arriving in the time interval $(-T, m)$. This intuition points out one key difference between the decay rates of $W(k)$ under SMART and LAS. While under LAS $\mathfrak{A}_{>k}(\vec{y})$ characterizes the effect of jobs with size larger than k, there is no such term in the decay rate of SMART (see (7)).

To illustrate the intuition of the above result, we will use the special case when jobs are only of sizes 1 and M. For LAS, in the special case when there only exist size 1 and M jobs, it follows from Theorem 2 that

$$\Pr(W^{(N)}(k) > m) \propto \max_{T\geq 0} \Pr(F_k^{(N)}(T) > C(T+m+1)) \tag{11}$$

where

$$F_k^{(N)}(T) = \begin{cases} A_1^N(-T,0), & k=1; \\ A_1^N(-T,m) + MA_M^N(-T,0) & \\ \quad +(M-1)A_M^N(1,m), & k=M. \end{cases}$$

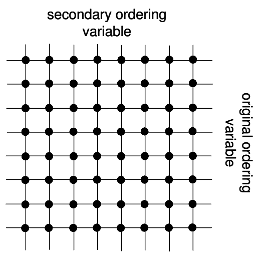


Figure 2: Two Dimensional Queueing Framework.

Note, that for size 1 jobs, SMART and LAS are asymptotically identical (i.e., the decay rates are the same). On the other-hand, for size M jobs, observe that for each fixed T, $F_M^{(N)}(T) \geq E_M^{(N)}(T)$, which immediately implies the delay of a job of original size M with LAS stochastically dominates the corresponding delay with SMART. Thus, the delay experienced by a size M job under LAS is larger than that under SMART (in distribution). Thus it follows that for the case where arriving jobs are one of two sizes (1 or M), LAS is uniformly worse than any SMART policy. We can prove the following corollary using a simple extension of the above example. The proof is omitted for space.

COROLLARY 3. *Any SMART policy is uniformly better (for any job size) than the LAS policy with respect to delay in the many sources large deviation regime, i.e.*

$$I_{\overline{W}}(k,m) \geq I_{\hat{W}}(k,m),$$

where $I_{\overline{W}}(k,m)$ and $I_{\hat{W}}(k,m)$ are the delay decay rates of all SMART policies and LAS respectively.

In Section 7, we will investigate the magnitude of the difference between LAS and SMART as a function of the system load and the job size distribution.

4. TWO DIMENSIONAL QUEUEING FRAMEWORK

In order to analyze LAS and the class of SMART policies in the many sources regime, we develop a new analysis framework that we will refer to as the *two dimensional queueing framework*. The two dimensional queueing framework is composed of two concepts: *discretization* and *ordering*.

First, by *discretization* we reduce the problem to considering only finite-sized jobs that are restricted to multiples of a unit size. More importantly, we assume that the server services the jobs in *discrete* amounts. For example, in this paper, we assume that the set of possible job sizes is $\mathcal{M} = \{1, 2, 3, \ldots, M\}$ and that the service occurs in unit size increments. The important consequence of discretization is that at any time slot, the distribution of the remaining size of a job is *discrete* and finite. This results in a more tractable description of the queue state.

The second important concept of the two dimensional queueing framework is *ordering*. Many scheduling policies specify a scheme of ordering (prioritization), in which the server considers some jobs more important and services those first. FCFS orders jobs by their time of arrival, PSJF by the job size (i.e., service requirement), SRPT by the remaining size (i.e., remaining service requirement), and LAS by the age (i.e., attained service). The two dimensional queueing framework takes this concept of ordering one step further by assigning a secondary ordering scheme to the previously existing one. In other words, jobs are serviced in the order specified by the scheduling policy, but when there are multiple jobs with the same priority, the secondary ordering scheme is used to select the next job to serve. For example, let us consider the ordering of "smaller job size first" as the secondary ordering for FCFS. In this case, when multiple jobs arrive to the system at the same time slot, smaller jobs are served before larger jobs. The importance of the secondary ordering is that it further constrains the policy, thus making the analysis more tractable, as we will see in the cases of SMART and LAS.

We note that by applying discretization and ordering to LAS and the policies in the SMART class, we *are not altering the performance of these policies in the asymptotic framework*. The discretization and ordering are simply modeling aids. In particular, we will use job size as a secondary ordering variable for LAS, but this does not mean that LAS needs to know the size of a job, rather that using the job size makes the analysis of LAS more tractable.

The term "two dimensional queueing framework" comes from the observation that the state of all jobs in the system at any time can be viewed as a two dimensional grid, where the y-axis is the variable related to the original ordering and the x-axis is the variable of the secondary ordering as shown in Figure 2. This setup allows the scheduling policy to be analyzed as a two dimensional priority queueing system.

5. SMART

We will now formally describe the SMART class and then prove that the decay rate for all SMART policies matches that for SRPT.

Formally, SMART is defined as follows [26]. Denote jobs using a, b, and c where job a has original size s_a and remaining size r_a. SMART is defined to be the set of scheduling policies that obey the following properties.

(i) **Bias Property:** *If $r_b > s_a$, then job a has priority over job b.*

(ii) **Consistency Property:** *If job a ever receives service while job b is in the system, thereafter job a has priority over job b.*

(iii) **Transitivity Property:** *If an arriving job b preempts job c; thereafter, until job c receives service, every arrival, a, with size $s_a < s_b$ is given priority over job c.*

Notice that the Bias Property guarantees that SMART policies favor "small" jobs, while the Consistency and Transitivity Properties enforce "coherency" in how priority is assigned over time.

The SMART class includes, for example, SRPT and PSJF. Furthermore, it is easy to prove that the SMART class includes all policies that assign to a job the product of its remaining size raised to the ith power and its original size raised to the jth power (for $i, j > 0$) and give priority to the job with lowest product. Such policies can provide lower *weighted* response times in many cases than SRPT. Further, the SMART class also includes a range of policies with more complicated priority schemes, even ones that do not maintain a static priority structure, e.g., a SMART policy may switch from using the SRPT rule to using the PSJF rule over time. Despite its breadth, many policies are excluded from SMART e.g. LAS and FCFS.

The SMART classification represents an emerging style of research based on analyzing large groups of policies instead of individual disciplines in an attempt to add structure to the space of scheduling policies that cannot be obtained through the analysis of individual policies. Beyond this theoretical motivation for studying classifications, the study of such classifications is important in practice because system designers can never implement the idealized policies (such as SRPT) that are the focus of theoretical research. By analyzing classifications of policies, the hope is that

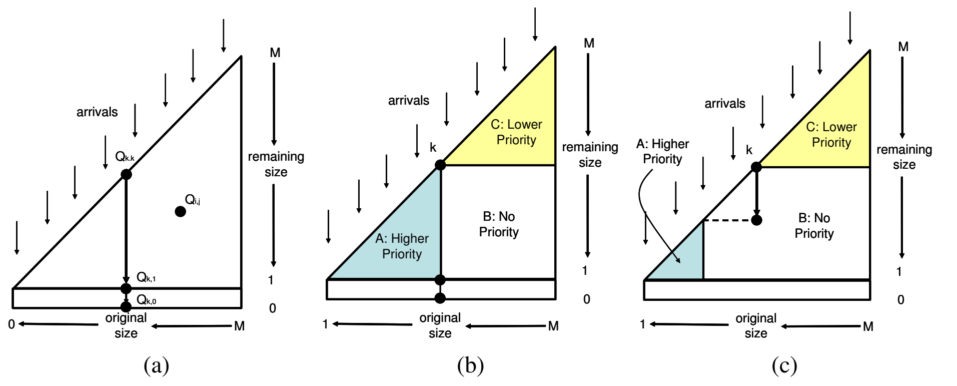


Figure 3: Illustrations of the two dimensional queueing framework for SMART policies. The progression of a job between queues while in the system is illustrated in (a). The priority structure for an incoming job is shown in (b) and for a partially served job is shown in (c).

theoretical results can be obtained for the unique, hybrid policies that are actually implemented.

Note that SMART falls into the two dimensional queueing framework very nicely. We arrange queues in a two dimensional grid where the x-axis is the original size of the job and the y-axis is the remaining size of the job (see Figure 3). A new job arrives to the queues on the upper most strip, where the original size and the remaining size are equal. The job then progresses through the system by moving downward until the remaining size becomes 0. We denote by $Q_{i,j}$, $i \geq j$, the queue that contains all the jobs that were originally of size i and have remaining size j. A job of size k arrives at $Q_{k,k}$, and then moves through $Q_{k,k-1}$, $Q_{k,k-2}$, ..., $Q_{k,1}, Q_{k,0}$ as service is received. We denote $Q_{i,j}(t)$ as the volume of $Q_{i,j}$ during time slot t. Additionally, we define Q_i as the queue containing all jobs of original size i and $Q_i(t) = \sum_{j=1}^{i} Q_{i,j}(t)$ similarly.

Within the two dimensional queueing framework, the properties in the definition of SMART specify a partial priority ordering on the queues. For instance, the Consistency Property guarantees that a job which has begun service can only be preempted by a new arrival of jobs. Further, when considering a new arrival of a tagged job, the Bias Property guarantees that all jobs with smaller remaining size will have higher priority. However, the Bias Property does not specify any ordering for jobs with equal original size or equal remaining processing time. The spirit of the two dimensional queueing framework is to provide a specific ordering in such situations, thus we adjust the Bias Property as follows.

(i) **Bias Property (2DQ):** *If $r_b \geq s_a$, then job a has priority over job b.*

This adjustment guarantees that, among jobs with equal original size, jobs with smaller remaining size are given higher priority, and among jobs with equal remaining size, jobs with smaller original size are assigned higher priority. Lastly, for jobs that have the same original size and have the same remaining processing time, the jobs are serviced using FCFS. Note that this change to the Bias Property does not alter the performance of SMART in the asymptotic framework, it is simply a modeling decision used to make the analysis tractable.

Let us now consider the behavior of a size k job under a SMART policy. Upon arrival, the job resides in $Q_{k,k}$. By the Bias Property (2DQ), the following queues have higher and lower priority compared to $Q_{k,k}$:

$$\text{Higher Priority: } \bigcup_{i=1}^{k} \bigcup_{j=1}^{k-1} Q_{i,j}, \quad \text{Lower Priority: } \bigcup_{i=k+1}^{M} \bigcup_{j=k}^{M} Q_{i,j} \tag{12}$$

This is illustrated in Figure 3. We denote the group of higher priority queues as area A and lower priority queues as area C. Note that, there is an another area where the queues do not have any fixed priority order compared to $Q_{k,k}$. Jobs in this area, B, can be serviced before or after the size k job. Due to the priority scheme, area A must be empty for the job in $Q_{k,k}$ to receive one unit of service. As depicted in Figure 3, when a job receives service it moves down the vertical line in the two dimensional queue and the priority areas change according to the position of the queue in which the job resides. For the partially serviced size k job to receive service, all queues in the corresponding area A, as depicted in Figure 3, must be empty. In both the cases of an incoming size k job and partially serviced size k job, the relative priority of queues in area B is unknown. However, the volume of jobs in area B can be bounded by the following observation, which will be used in the derivation of the decay rate. The Consistency Property guarantees that a job in service can be preempted only by a new arrival. Thus at every time slot, at most one additional job can become partially serviced.

5.1 The decay rate of delay under SMART

We now derive the actual delay decay rate of SMART in the many sources regime. We prove Theorem 1 by first considering the virtual delay, $\Pr(\overline{V}^{(N)}(k) > m)$, for size k jobs then deriving the actual delay, $\Pr(\overline{W}^{(N)}(k) > m)$.

Define $I_{\overline{W}}(k, m)$ as the decay rate of the actual delay of a size k job under SMART with total service rate NC. Let $I_{\overline{V}_\mu}(k, m)$ denote the decay rate of the virtual delay of a size k job under SMART with total service rate $N\mu$. For bounding purposes, we consider the virtual delay of SMART where an additional size k job is inserted before the virtual job, and denote it as $I_{\tilde{V}_\mu}(k, m)$.

LEMMA 4. *For any $k \in \mathcal{M}$, under any SMART policy*

$$I_{\tilde{V}_C}(k, m) \leq I_{\overline{W}}(k, m) \leq I_{\overline{V}_C}(k, m) \tag{13}$$

PROOF. First, we derive the upper bound by showing

$$\Pr\left(\overline{V}^{(N)}(k) > m | A^N(0,0) > 0\right) \leq \Pr\left(\overline{W}^{(N)}(k) > m\right) \tag{14}$$

Note that a virtual job (with size 0) need only to arrive at the front of the queue to be fully serviced. Thus, (14) follows from the observation that, if a fictitious job did not leave the system (i.e. arrive at the head of the queue) before time m, then the actual job did not leave the queue. That is, the actual job did not even receive one unit of service. Thus, the actual job is guaranteed to have not left the system by time m. This queue state is depicted in Figure 4, where the last job corresponds to the actual job. Thus, we have

$$\begin{aligned} & -\frac{1}{N}\log \Pr\left(\overline{W}^{(N)}(k) > m\right) \\ \leq \quad & -\frac{1}{N}\log \Pr\left(\overline{V}^{(N)}(k) > m | A^N(0,0) > 0\right) \qquad (15) \end{aligned}$$

As $N \to \infty$, (15) can be further upper bounded by $I_{\overline{V}_C}(k,m)$ using similar techniques to those in [24]. Finally,

$$-\frac{1}{N}\log \Pr\left(\overline{W}^{(N)}(k) > m\right) \underset{N\to\infty}{\longrightarrow} I_{\overline{W}}(k,m)$$

gives the upper bound.

To prove the lower bound, we add an extra size k job in front of the virtual queue and consider the virtual delay, $\tilde{V}^{(N)}(k)$. The queue state is depicted in Figure 4. We prove the following inequality using a contra-positive argument..

$$\Pr(\overline{W}^{(N)}(k) > m) \quad \leq \quad \Pr\left(\tilde{V}^{(N)}(k) > m | A^N(0,0) > 0\right) (16)$$

The event $\left\{\tilde{V}^{(N)}(k) \leq m | A^N(0,0) > 0\right\}$ is equivalent to the statement that the virtual job leaves the system before time m. Note that for a virtual job to leave the system, all that is required is for the virtual job to reach the head of the queue. The virtual job reaches the head of the queue when the extra job leaves the queue, i.e., the extra job gets one unit of service. The key observation is that, the extra job need not be fully serviced, only partially serviced. However, for the extra job to be even partially serviced, the last job of the batch arrival must leave the system completely. This is due to the secondary priority scheme introduced in the two dimensional queueing framework: jobs with the same original size but with smaller remaining sizes have higher priority. This last job in the front of the virtual job is the job that represents the actual delay. Thus, we have the following

$$\left\{\tilde{V}^{(N)}(k) \leq m | A^N(0,0) > 0\right\} \Rightarrow \left\{\overline{W}^{(N)}(k) \leq m\right\}$$

where $A \Rightarrow B$ denotes A implies B. This proves (16) by a contra-positive argument. Thus we have

$$\begin{aligned} & -\frac{1}{N}\log \Pr\left(\tilde{V}^{(N)}(k) > m | A^N(0,0) > 0\right) \\ \leq \quad & -\frac{1}{N}\log \Pr\left(\overline{W}^{(N)}(k) > m\right). \end{aligned}$$

As $N \to \infty$, above equation can be lower bounded by $I_{\tilde{V}_C}(k,m)$ using similar arguments to [24]. Finally, noting the definition of $I_{\overline{W}}(k,m)$, completes the proof. □

We are now ready to prove Theorem 1. Define $\overline{B}^N_{(k,k)}(a,b)$ as the volume of potential service that jobs in $Q_{k,k}$ can receive in interval (a,b) under SMART. Potential service corresponds to the maximum amount of service that can be received if the corresponding queue is never empty. Note that $Q_{k,k}$ does not include original size k jobs that have received partial service.

PROOF. *(of Theorem 1)* First, we derive the lower bound on the decay rate in (6) by finding an upper bound of $\Pr\left(\tilde{V}^{(N)}(k) > m\right)$ and using Lemma 4.

Let us consider the virtual delay in SMART with the extra job, i.e., $\Pr\left(\tilde{V}^{(N)}(k) > m\right)$. Observe that if the virtual delay with the extra size k job exceeds m, then we have that the queue length at time zero (i.e. $Q_{k,k}(0)$) and the extra job is not served by time m.

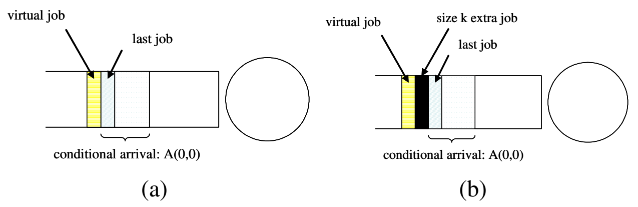


Figure 4: Depiction of the queue state for the lower bound (a) and upper bound (b) in the analysis of SMART.

In other words, $\left\{\tilde{V}^{(N)}(k) > m\right\} \Rightarrow \left\{Q_{k,k}(0) + k > \overline{B}^N_{(k,k)}(1,m)\right\}$, which results in

$$\Pr\left(\tilde{V}^{(N)}(k) > m\right) \leq \Pr\left(Q_{k,k}(0) + k > \overline{B}^N_{(k,k)}(1,m)\right) \qquad (17)$$

From Loynes' formula, we have

$$\begin{aligned} & \Pr\left(Q_{k,k}(0) + k > \overline{B}^N_{(k,k)}(1,m)\right) \\ = \quad & \Pr\left(\sup_{T\geq 0}\left[kA^N_k(-T,0) + k - \overline{B}^N_{(k,k)}(-T,m)\right] \geq 0\right) \\ = \quad & \Pr\left(kA^N_k(-T^*,0) - \overline{B}^N_{(k,k)}(-T^*,m) + k \geq 0\right). \quad (18) \end{aligned}$$

Note that in addition to the volume of $Q_{k,k}$ ($Q_{k,k}(0)$), k is added in deriving (18) due to the construction of $\tilde{V}$. Also, $-T^*$ is the most recent time in the past such that $Q_{k,k}(-T^* - 1) = 0$.

Now, we derive a lower bound on $\overline{B}^N_{(k,k)}(-T^*,m)$, which will in turn provide an upper bound of $\Pr(\tilde{V}^{(N)}(k) > m)$. We make use of the priority scheme of SMART in the two dimensional queueing framework, which has been discussed above. An observation was made that area A is higher priority compared to $Q_{k,k}$, and the queues in area B may or may not have higher priority. Thus a simple lower bound on $\hat{B}^N_{(k,k)}(-T^*,m)$ is the available service assuming that both areas A and B have higher priority. The volume of service that area A requires can be derived using the fact that all queues in area A are empty at time $-T^* - 1$, i.e., $Q_{(i,j)}(-T^* - 1) = 0$, for all $i \leq k$ and $j \leq k-1$. The proof follows immediately from Theorem 4.1 in [24], which is stated in the context of priority queues. Additionally, the volume of service that area B requires during interval $(-T^*, m)$ is upper bounded by $(T^* + m + 1)(M-1)$. This is due to the observation that at most a single partially serviced job can occur in a time slot, and the worst partially serviced job is of size $M-1$. Thus $\overline{B}^N_{(k,k)}(-T^*,m)$ is lower bounded as follows.

$$\begin{aligned} \overline{B}^N_{(k,k)}(-T^*,m) \quad \geq \quad & NC(T^*+m+1) - \sum_{i=1}^{k}\sum_{j=1}^{k-1} Q_{(i,j)}(-T^*-1) \\ & - \sum_{i=1}^{k-1} iA^N_i(-T^*,m) - (T^*+m+1)(M-1) \\ = \quad & NC(T^*+m+1) - \sum_{i=1}^{k-1} iA^N_i(-T^*,m) \\ & -(T^*+m+1)(M-1). \qquad (19) \end{aligned}$$

From (17), (18), and (19), we have

$$\begin{aligned} & \Pr\left(\tilde{V}^{(N)}(k) > m\right) \\ \leq \quad & \Pr\left[kA^N_k(-T^*,0) + k - \overline{B}^N_{(k,k)}(-T^*,m) > 0\right] \\ \leq \quad & \Pr\left[kA^N_k(-T^*,0) + \sum_{i=1}^{k-1} iA^N_i(-T^*,m) \right. \\ & \quad \left. -NC(T^*+m+1) + (T^*+m+1)(M-1) + k > 0\right] \\ \leq \quad & \Pr\left[kA^N_k(-T^*,0) + \sum_{i=1}^{k-1} iA^N_i(-T^*,m) - NC(T^*+m+1)\right. \end{aligned}$$

$$
\begin{aligned}
& +2(T^*+m+1)M > 0] \\
\leq\ & \Pr\left[kA_k^N(-T^*,0) + \sum_{i=1}^{k-1} iA_i^N(-T^*,m) \right. \\
& \left. -N\left(C - \frac{2M}{N}\right)(T^*+m+1) > 0\right] \\
\leq\ & \Pr\left[\bigcup_{T\geq 0}\left(kA_k^N(-T,0) + \sum_{i=1}^{k-1} iA_i^N(-T,m)\right.\right. \\
& \left.\left. -N\left(C - \frac{2M}{N}\right)(T+m+1) > 0\right)\right] \\
\leq\ & \sum_{T\geq 0}\Pr\left[kA_k^N(-T,0) + \sum_{i=1}^{k-1} iA_i^N(-T,m)\right. \\
& \left. -N\left(C - \frac{2M}{N}\right)(T+m+1) > 0\right],
\end{aligned}
$$

Fix any $\epsilon > 0$ and observe that for N large enough, we have $(C - \frac{2M}{N}) > (C - \epsilon)$. Hence

$$
\begin{aligned}
\Pr\left(\tilde{V}^{(N)}(k) > m\right) \quad \leq \quad & \sum_{T\geq 0}\Pr\left[kA_k^N(-T,0) + \sum_{i=1}^{k-1} iA_i^N(-T,m)\right. \\
& \left. -N(C-\epsilon)(T+m+1) > 0\right]. \qquad (20)
\end{aligned}
$$

Note that (20) is the expression for the decay rate of size k jobs in PRI having capacity $C - \epsilon$, which was described in Section 3. Using similar techniques as in [7, 11], it follows that the lower bound of $I_{\tilde{V}_\mu}(k,m)$ is $I_{\overline{V}_{C-\epsilon}}(k,m)$. Applying Lemma 4, we have the lower bound on the decay rate in (6). Specifically, we apply the contraction principle, to obtain the closed form expression of $I_{V_{C-\epsilon}}(k,m)$.

Next, we derive the upper bound on the decay rate in (6) by deriving a lower bound on $\Pr(V^{(N)}(k) > m)$ and combining it with the result of Lemma 4. We do so by comparing the SMART policy with a priority queueing system which lower bounds the delay experienced by the job.

Consider a PRI(see Section 3) system with capacity NC, which we describe again. This system consists of M queues, with jobs of original size k arriving to queue-k. Partially served jobs in this system continue to reside in the same queue and the jobs in each queue are served in a FCFS manner. In comparing the PRI system to SMART, we can think of PRI as a SMART scheme where $Q_k = \sum_{i=1}^{k} Q_{(k,i)}$ and priorities are assigned such that area A has higher priority whereas area B and C have lower priority.

Denote $V^{(N)}(k)$ as the virtual delay of a size k job for PRI. Then event $\{V^{(N)}(k) \leq m\}$ of PRI ensures the event $\{\overline{V}^{(N)}(k) \leq m\}$ of SMART policies. This comes from the fact that the external arrival to both PRI and SMART are the same but the residual service available to $Q_{k,k}$ in SMART is upper bounded by the residual service for queue-k of PRI. This follows from the fact that the residual service of SMART is the remaining service after servicing of all of area A and possibly a part or all of area B. However, the residual service in PRI is that of after servicing only the area A, and none of area B. Thus PRI provides a lower bound on the virtual delay of a job compared to SMART. In other words, we have $I_{\overline{V}_C}(k,m) \leq I_V(k,m)$ and combining it with Lemma 4 the proof is complete. $\square$

6. LEAST ATTAINED SERVICE

In this section we will derive the decay rate for delay under LAS using the two dimensional queueing framework. This may seem surprising since the prioritization of LAS depends only on the attained service of a job; however adding a secondary variable is the key to making the analysis tractable.

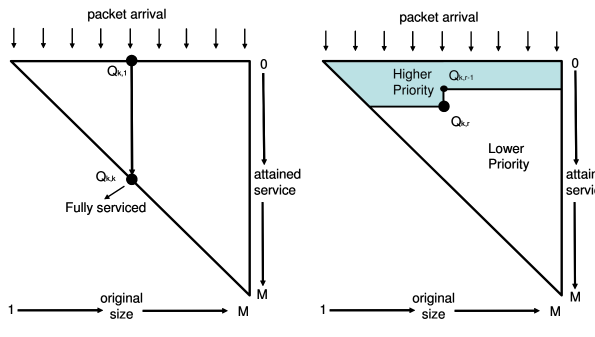


Figure 5: Illustrations of the two dimensional queueing framework for LAS. The progression of a job between queues while in the system is illustrated in (a). The priority structure for a job is shown in (b).

The secondary variable we add is the original job size. This means that, in the event of ties in attained service, instead of sharing the server among the jobs with equal attained service, the job with the smallest original size is served first. Further, jobs that have the same original size and the same attained service are serviced according to a FCFS rule. Keep in mind that jobs are not served with the full service capacity, but are only serviced a single unit at once, which is consistent with the discrete version of PS. Note that using the job size as a secondary ordering variable does not alter the performance of LAS in the asymptotic framework, it is simply a modeling decision used to make the analysis tractable.

Formally, a queue $Q_{i,j},\ i \geq j$, denotes the queue that contains all the jobs having original size i that have received j unit of service. Thus a job of original size k arrives to $Q_{k,0}$ and then progresses to $Q_{k,1}$, $Q_{k,2}$ ... $Q_{k,k-1}, Q_{k,k}$, at which point the job is fully serviced and leaves the system. This is depicted in Figure 5. We again denote $Q_{i,j}(t)$ as the volume of $Q_{i,j}$ at time t, and $Q_i(t)$ as the volume of the queue that contains all jobs that have original size i, Q_i, where $Q_i = \bigcup_{j=0}^{j=i-1} Q_{i,j}$.

Let us now consider a tagged size k job that arrives in the system at time 0 and has received r units of service (see Figure 5). By the definition of LAS, all jobs that have been served less than r units have higher priority than the tagged job. Additionally, jobs with smaller original size that have attained service r also have higher priority. In other words, for any job in $Q_{k,r}$ to be serviced an additional unit, the following queues must be empty,

$$
\bigcup_{i=1}^{k-1}\bigcup_{j=0}^{r} Q_{i,j} + \bigcup_{i=k+1}^{M}\bigcup_{j=0}^{r-1} Q_{i,j} + \bigcup_{j=0}^{r-1} Q_{k,j}
$$

Further, since in each queue, $Q_{i,j}$, jobs are serviced in a FCFS order, jobs that arrived to Q_k before the tagged job must be serviced r units and job that arrived after must be served only $r-1$ units. The same argument can be made for $Q_{k,k-1}$, which is the queue for jobs that will leave the system once it is served again.

6.1 Bounding the decay rate of virtual delay under LAS

We start the analysis of LAS by deriving a bound on the virtual delay under LAS. Denote the virtual delay of LAS as $\hat{V}(0)$. We bound the probability that the virtual delay for size k jobs exceeds m, $\Pr(\hat{V}^{(N)}(k) > m)$, as follows.

LEMMA 5. *For any $k \in \mathcal{M}$, the virtual delay of size k jobs in*

LAS satisfies

$$\begin{aligned}
& \Pr\left(\hat{V}^{(N)}(k) > m\right) \\
\leq\ & \Pr\left(\sup_{T\geq 0}\left(kA_k^N(-T,0) + (k-1)A_k^N(1,m)\right.\right. \\
& \qquad \left.\left. -\hat{B}_k^N(-T,m)\right) > 0\right) \\
=\ & \Pr\left(kA_k^N(-T^*,0) + (k-1)A_k^N(1,m)\right. \\
& \qquad \left. -\hat{B}_k^N(-T^*,m) > 0\right), \qquad (21)
\end{aligned}$$

where $A_k^N(-T,0)$ is the number of size k job arrivals in the interval $(-T,0)$, $\hat{B}_k(-T,m)$ is the service available to Q_k during $(-T,m)$, and T^* is the last time before 0 that $Q_k(-T^*-1)=0$

The theorem states that a *possible* scenario in which the event $\{\hat{V}^{(N)}(k) > m\}$ could occur is when the total volume of arrivals for size k job before the virtual job and a portion $((k-1)/k)$ of the volume of arrivals for size k job after the virtual job, exceed the available capacity for Q_k .

PROOF. Consider Q_k. By the operation of LAS, $\{\hat{V}^{(N)}(k) > m\}$ implies that all jobs in $Q_k(0)$ have not been fully serviced by time m, i.e., $\{kA_k^N(-T^*,0) > \hat{B}_k^N(-T^*,m)\}$. Since adding more jobs makes the event of exceeding the available capacity more probable, we have the following.

$$\begin{aligned}
& \Pr\left(\hat{V}^{(N)}(k) > m\right) \\
\leq\ & \Pr\left(\sup_{T\geq 0}\left(kA_k^N(-T,0) - \hat{B}_k^N(-T,0)\right) - \hat{B}_k^N(1,m) > 0\right) \\
=\ & \Pr\left(\sup_{T\geq 0}\left(kA_k^N(-T,0) - \hat{B}_k^N(-T,m)\right) > 0\right) \\
\leq\ & \Pr\left(\sup_{T\geq 0}\left(kA_k^N(-T,0) + (k-1)A_k^N(1,m)\right.\right. \\
& \qquad \left.\left. -\hat{B}_k^N(-T,m)\right) > 0\right)
\end{aligned}$$

Finally, (21) follows from Loynes' formula. □

Note that the total available capacity to Q_k in interval $(-T,m)$ with regards to the event $\{\hat{V}^{(N)}(k) > m\}$, i.e. $\hat{B}_k^N(-T,m)$, is the remaining capacity after all the higher priority queues are served in LAS. So, the following holds

$$\begin{aligned}
& \hat{B}_k^N(-T,m) \\
\geq\ & NC(T+m+1) - \sum_{i=1}^{k-1} iA_i^N(-T,m) - \sum_{i=1}^{k-1} Q_i(-T-1) \\
& - \sum_{i=k+1}^{M}(k-1)A_i^N(-T,m) - \sum_{i=k+1}^{M}\sum_{j=0}^{k-2} Q_{i,j}(-T-1)
\end{aligned}$$
(22)

Using Lemma 5 and the above, we can derive the following.

THEOREM 6. *The virtual decay rate of size k jobs under LAS is bounded as follows*

$$I_{\hat{V}}(k,m) \geq \inf_{T\geq 0}\left[\inf_{\vec{y}:\mathcal{Y}}\left\{\inf_{\vec{x}:\mathcal{X}}\left(\mathfrak{A}_{<k}(\vec{y}) + \mathfrak{A}_k(\vec{y}) + \mathfrak{A}_{>k}(\vec{y})\right)\right\}\right], \quad (23)$$

where $\mathcal{Y}$, $\mathcal{X}$, $\mathfrak{A}_{<k}(\vec{y})$, $\mathfrak{A}_k(\vec{y})$, and $\mathfrak{A}_{>k}(\vec{y})$ are defined as in Theorem 2.

PROOF. Combining (22) with Lemma 5 we have the following upper bound on the probability of $\{\hat{V}^{(N)}(k) > m\}$.

$$\begin{aligned}
& \Pr\left(\hat{V}^{(N)}(k) > m\right) \\
\leq\ & \Pr\left(\sup_{T\geq 0}\left(kA_k^N(-T,0) + (k-1)A_k^N(1,m)\right.\right. \\
& \qquad \left.\left. -\hat{B}_k^N(-T,m)\right) > 0\right) \\
=\ & \Pr\left(kA_k^N(-T^*,0) + (k-1)A_k^N(1,m) - \hat{B}_k^N(-T^*,m) > 0\right) \\
\leq\ & \Pr\left(kA_k^N(-T^*,0) + (k-1)A_k^N(1,m) + \sum_{i=1}^{k-1} iA_i^N(-T^*,m) + \right. \\
& \sum_{i=1}^{k-1} Q_i(-T^*-1) + \sum_{i=k+1}^{M}(k-1)A_i^N(-T^*,m) \\
& \left. + \sum_{i=k+1}^{M}\sum_{j=0}^{k-2} Q_{i,j}(-T^*-1) - NC(T^*+m+1) > 0\right) \\
=\ & \Pr\left(kA_k^N(-T^*,0) + (k-1)A_k^N(1,m) + \sum_{i=1}^{k-1} iA_i^N(-T^*,m)\right. \\
& \left. + \sum_{i=k+1}^{M}(k-1)A_i^N(-T^*,m) - NC(T^*+m+1) > 0\right) \quad (24) \\
\leq\ & \Pr\left(\bigcup_{T\geq 0}\left(kA_k^N(-T,0) + (k-1)A_k^N(1,m) + \sum_{i=1}^{k-1} iA_i^N(-T,m)\right.\right. \\
& \left.\left. + \sum_{i=k+1}^{M}(k-1)A_i^N(-T,m) - NC(T+m+1)\right) > 0\right) \\
\leq\ & \sum_{T\geq 0}\Pr\left(kA_k^N(-T,0) + (k-1)A_k^N(1,m) + \sum_{i=1}^{k-1} iA_i^N(-T,m)\right. \\
& \left. + \sum_{i=k+1}^{M}(k-1)A_i^N(-T,m) - NC(T+m+1) > 0\right)
\end{aligned}$$

where (24) follows from the following observation that $Q_i(-T^*-1) = 0$ for $1 \leq i \leq k-1$ and $Q_{i,j}(-T^*-1) = 0$ for $k+1 \leq i \leq M$, $0 \leq j \leq k-2$. The justification is similar to that for a priority queueing system. Namely, since the above queues have higher priority than Q_k, when Q_k is empty all higher priority queues must be empty. Applying the contraction principle completes the proof. □

Theorem 6 provides a *possible* scenario in which $\{\hat{V}^{(N)}(k) > m\}$ could occur. To show that this scenario is indeed the most dominant and controls the behavior of the probability, in the next section we will show that the same scenario provides the upper bound on the decay rate of $\{\hat{V}^{(N)}(k) > m\}$.

6.2 The decay rate of delay under LAS

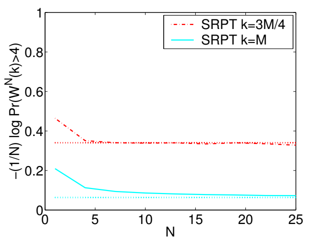
We now develop an upper bound for the decay rate of LAS which is tight with the lower bound derived in Section 6.1, i.e., we develop the tight lower bound for the tail probability. The main argument for the upper bound is that, using the two dimensional queueing framework, LAS can be viewed as a simple priority queueing system, as was the case with SMART. Thus, using a similar analysis, we can derive the tight upper bound for the decay rate.

THEOREM 7. *The probability of the virtual delay for size k jobs in LAS can be lower bounded as*

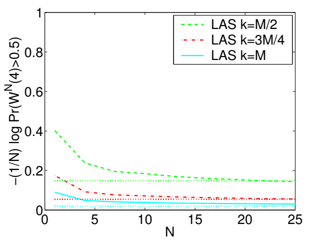
$$\begin{aligned}
& \Pr\left(\hat{V}^{(N)}(k) > m\right) \geq \Pr\left(\inf_{0\leq l\leq m}\left\{\sum_{i=1}^{k-1} iA_i^N(-T^*,l)\right.\right. \\
+\ & kA_k^N(-T^*,0) + (k-1)A_k^N(1,l) \\
+\ & \left.\left.\sum_{i=k+1}^{M}(k-1)A_i^N(-T^*,l) - NC(T^*+l+1)\right\} > 0\right) (25)
\end{aligned}$$

where $-T^*$ is the last time before time 0 when $Q_k(-T^*-1) = 0$.

PROOF. As explained in Section 6.1, there exists a group of queues and arrivals that constitute higher priority compared to the virtual job that arrived at time 0. At time l the volume of jobs from

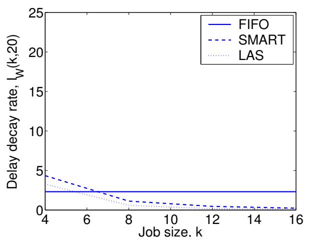


(a) SRPT

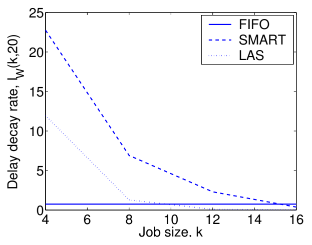


(b) LAS

Figure 6: Plot of the rate of convergence to the decay rate under the uniform workload with $M = 16$, $\rho = 0.8$, and $m = 4$. The asymptotic decay rate is shown as a dotted line. Note that only the decay rates of the larger sizes are shown because only these can be estimated accurately in simulation since a large delay for smaller job sizes is a very low probability event as N grows. Though not shown here, we have found similar convergence rates under other SMART policies.



(a) Power-law



(b) High variability

Figure 7: Plot of the decay rate as a function of the job size, k, with the threshold $m = 20$ and maximum job size $M = 16$ held fixed. The load is 0.8. Recall that $I_W(k, m)$ measures the decay rate of $Pr(W(k) > m)$ and that a larger $I_W(k, m)$ indicates a stochastically smaller delay. Note that since decay rates of size 1 jobs for both SMART and LAS are infinite, they are omitted.

higher priority queues and arrivals is

$$\sum_{i=1}^{k-1}\sum_{j=0}^{i-1} Q_{i,j}(l) + \sum_{i=k+1}^{M}\sum_{j=0}^{k-2} Q_{i,j}(l) + \sum_{j=0}^{k-2} Q_{k,j}(l) + A_k(-T^*,0). \tag{26}$$

If the higher priority queues and arrivals with respect to the virtual job (26) are never empty at any time during $(0, m)$, then the virtual job is guaranteed not to leave the system in the interval $(0, m)$. Based on this observation we derive a lower bound on $\Pr(\hat{V}^{(N)}(k) > m)$ as follows.

$$\begin{aligned}
&\Pr\left(\hat{V}^{(N)}(k) > m\right) \\
\geq\ & \Pr\left(\inf_{0\leq l\leq m}\left\{\sum_{i=1}^{k-1}\sum_{j=0}^{i-1} Q_{i,j}(l) + \sum_{i=k+1}^{M}\sum_{j=0}^{k-2} Q_{i,j}(l)\right.\right. \\
&\quad \left.\left. + \sum_{j=0}^{k-1} Q_{k,j}(l) + A_k(-T^*,0)\right\} > 0\right) \\
=\ & \Pr\left(\inf_{0\leq l\leq m}\left\{\sum_{i=1}^{k-1}\sum_{j=0}^{i-1} Q_{i,j}(-T^*-1) + \sum_{i=1}^{k-1} iA_i^N(-T^*,l)\right.\right. \\
&\quad + \sum_{i=k+1}^{M}\sum_{j=0}^{k-1} Q_{i,j}(-T^*-1) + \sum_{i=k+1}^{M}(k-1)A_i^N(-T^*,l) \\
&\quad \left.\left. + kA_k^N(-T^*,0) + (k-1)A_k^N(1,l) - NC(T^*+l+1)\right\} > 0\right) \\
=\ & \Pr\left(\inf_{0\leq l\leq m}\left\{\sum_{i=1}^{k-1} iA_i^N(-T^*,l) + \sum_{i=k+1}^{M}(k-1)A_i^N(-T^*,l)\right.\right. \\
&\quad \left.\left. + kA_k^N(-T^*,0) + (k-1)A_k^N(1,l) - NC(T^*+l+1)\right\} > 0\right)
\end{aligned}$$

In the calculation above, note that we have argued in Section 6.1 that at time $-T^* - 1$ all higher priority queues are empty. □

Extending the above to obtain a tight lower bound on the decay rate of LAS is difficult without further assumptions on the inputs. We make 2 assumptions in order to complete the derivation.

ASSUMPTION 1. *Let A_{high} denote the sum of all higher priority arrivals described in Theorem 7, then we assume that the corresponding rate function satisfies*

$$I_{A_{high}}^{(-T^*,l)}\left(C(T^*+l+1) - v\right) < I_{A_{high}}^{(-T^*,0)}\left(C(T^*+1)\right) \tag{27}$$

for $v \in [v^* - \delta, v^* + \delta]$, $v^* > 0$, and $\delta > 0$ sufficiently small.

ASSUMPTION 2. *Define*

$$\vec{A}_{high} = \left(\ldots, A_{high}^N(-T,0), \ldots, A_{high}^N(-1,0), A_{high}^N(0,0)\right)$$

Then, we assume that the stochastic process $\vec{A}_{high}$ satisfies

$$\left(\vec{A}_{high}|A_{high}^N(0,0) = 0\right) \leq_{st} \left(\vec{A}_{high}|A_{high}^N(0,0) > 0\right).$$

where $\vec{A}_{high}$ was defined in Assumption 1.

The first assumption is equivalent to the decay rate being additive. Intuitively, a decay rate with the property of additive functionals implies that the occurrence of a rare event in the large deviation framework happens in a straight line. This assumption has been used extensively in large deviation literature [2, 13, 8]. Further, arrival processes that satisfy Assumption 1 include many common processes such as all stationary and Markov dependent processes. Additionally, if the arrival process is of Levy type then the decay rate of the the arrival satisfies (27).

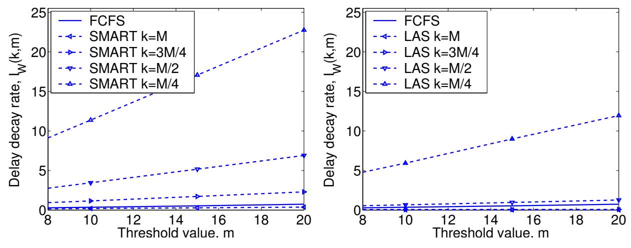
The second assumption allows us to show that the virtual delay decay rate is equal to the actual delay decay rate. This assumption essentially says that the arrival process has the property that if there are very few arrivals in a given time slot, there were also very few arrivals in the immediate past (and vice versa). It is a kind of "burstiness" assumption for the source.

We are now ready to complete the proof of Theorem 2.

PROOF. *(of Theorem 2)* It follows from Theorem 7 and Assumption 1 that $I_{\hat{V}}(k, m)$ is upper bounded by the expression in (10) using the same technique as in [8]. Further, applying Theorem 6, we obtain equality. Lastly, using similar arguments as in [24], we can conclude that the actual delay decay rate is equal to the virtual delay decay rate under Assumption 2, which completes the proof. □

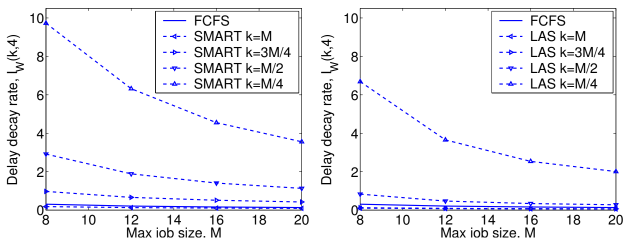
7. NUMERICAL EXPERIMENTS

Though Theorems 1 and 2 provide expressions for the decay rates of SMART policies and LAS in the many sources regime, the complicated nature of these formulas hide the behavior of the decay rates. In this section, we will use simulations and numerical experiments to illustrate how the decay rate $I_W(k, m)$, and thus $Pr(W(k) > m)$, of SMART and LAS are affected by the load (ρ), the variability of the service distribution, the range of the service distribution (M), and the threshold value (m). In performing these studies we focus on three practical questions:



(a) SMART (b) LAS

Figure 8: Plot of the decay rate as a function of the threshold m under the high variability workload under both SMART and LAS with the maximum job size $M = 16$ and $\rho = 0.8$. Each line in the figures corresponds to the decay rate of delay experienced by a specific job size k. The decay rate of FCFS is included as a benchmark. Note that since decay rates of size 1 jobs are infinite, they are omitted.



(a) SMART (b) LAS

Figure 9: Plot of the decay rate as a function of the maximum job size M under the high variability workload under both SMART and LAS with the threshold $m = 4$ and $\rho = 0.8$. Each line in the figures corresponds to the decay rate of delay experienced by a specific job size k. The decay rate of FCFS is included as a benchmark. Note that since decay rates of size 1 jobs are infinite, they are omitted.

1. How does the delay distribution behave under a large, but not infinite, number of flows? That is, what is the rate of convergence to the decay rate, $I_W(k, m)$?
2. How does the decay rate for a job of size k vary across k? That is, how much do large job sizes suffer under policies that bias towards small job sizes?
3. How much penalty does LAS pay for not using job size information to prioritize? That is, by how much does SMART outperform LAS?

7.1 Setup

In our experiments, we assume that jobs are of sizes 1, $M/4$, $M/2$, $3M/4$, or M, and we vary M between 8 and 20. Each job size arrives according to an on-off process, i.e. in each discrete interval a size k job arrives with probability p_k. We assume that the capacity of the system, C, is 1.

We will consider three cases for the distribution of job sizes, which we refer to as *uniform*, *power-law*, and *high variability*. In the *uniform* case each job size is equally likely. In the *power-law* case, the arrival probabilities follow the power-law distribution with exponent 2, i.e. a discrete and truncated counterpart of the Pareto distribution. Note that due to the small spread of job sizes, this distribution is not highly variable; thus, to study the impact of variability, we also consider a distribution where the largest jobs make up half the load (as has been observed in web file sizes). In particular, the *high variability* workload has size 1, $M/4$, $M/2$, $3M/4$, and M job arrivals make up $1/24$, $1/12$, $1/8$, $1/4$, and $1/2$ of the total load.

7.2 Discussion

Moving to our results, we will first investigate question 1. Figure 6 illustrates the convergence of the delay distribution to the asymptotic decay rate as N grows under both SRPT and LAS. The dotted lines are the numeric calculations of the asymptotic decay rates proven in the paper and the other lines are generated using an event-driven simulation. The simulation matches the uniform workload described above except that a Poisson arrival process is used. Thus, in addition to the error from using a finite N, Figure 6 illustrates the error from the discretization of arrivals. Note that when $N = 20$ there is already little difference between the empirical decay rate and the asymptotic limit under either SRPT or LAS. Further, though omitted for lack of space, we have also generated plots that indicating similar convergence behavior under two other SMART policies: PSJF and RS (which prioritizes towards the smallest product of remaining size and original size). Thus, in answer to question 1, it seems that rate of convergence to the decay rate is very fast, and thus the asymptotic decay rate provides information that is useful in practical settings such as high traffic web servers and routers where there are far more than 20 simultaneous flows.

To address question 2, Figure 7 illustrates how the decay rate for a job of size k varies across k under SMART policies, LAS, and FCFS (which we include as a baseline for comparison). The results are shown for both the power-law and high variability workloads under high load. *Note that a larger decay rate indicates a stochastically smaller delay.*

The first observation we make is that, in each of the plots, small job sizes have much better decay rates under SMART policies and LAS than under FCFS; whereas large job sizes have better decay rates under FCFS than under LAS and SMART. Thus, there is always crossover point for each of SMART and LAS where their decay rate "crosses over" that of FCFS. Figure 7 illustrates that the crossover point is highly dependent on the service distribution. When the load is high and the largest jobs make up a significant fraction of the load, the decay rate of SMART does not cross that of FCFS until the largest job size, $k = M$. The behavior of the crossover point can be understood using the following key observation: while the decay rate under FCFS gets worse (smaller) as the load of largest jobs is increased and the total load is held constant, the decay rates of LAS and SMART get better (larger). Thus, in answer to question 2 above, since large job sizes make up a significant fraction of the load in many computer applications, it seems that one need not worry to much about the suffering of large job sizes under policies that prioritize small jobs. Though not shown, we have also investigated the impact of load on the crossover point and found that load only changes the magnitude of the decay rates (the higher the load, the lower the decay rate), not the relative behavior of the decay rates under FCFS, LAS, and SMART policies.

The next observation we make from Figure 7 is that both SMART and LAS exhibit a similar trend in decay rate across k, with SMART always providing stochastically smaller delays than LAS. Further, in answer to question 3, Figure 7 illustrates that the improvement of SMART over LAS is again highly dependent on the job size distribution: as the load of the largest jobs increases, the difference in the decay rates of SMART and LAS increases. The fact that SMART is better for small jobs follows from the operation of two policies: small jobs typically do not get preempted under SMART, but are preempted by all arrivals under LAS. However, the result that SMART is better than LAS even for larger jobs less obvious. An explanation for this fact is that under LAS, though larger jobs gain higher priority at arrival compared to SMART, as large jobs receive service their priority is dropping quickly under LAS but may be increasing under SMART.

In Figure 7, both the maximum job size M and the threshold value m are fixed. However, we have also investigated a range of other M and m and we illustrate the effect of these variables in Figures 8 and 9 respectively. Note that in both of these figures, we again see that the trends under SMART and LAS are parallel and that SMART always provides stochastically smaller delay than LAS. These plots also illustrate the effect of increasing M and m on the decay rate of delay. As the threshold value m increases, Figure 8 shows that the decay rate increases, and thus $Pr(W(k) > m)$ decreases. It is interesting to note that the decay rate seems to grow linearly with m for all jobs sizes k under both LAS and SMART. As the maximum job size M increases, Figure 9 shows that the decay rate of all job sizes decreases, which is not surprising since this leads to an increase in service times for all job sizes.

8. CONCLUSION

We conclude the paper by summarizing our results. We have derived expressions for the delay decay rate of the SMART class and LAS in the *many sources* large deviation framework. We have shown that all scheduling policies in the SMART class have the same decay rate. In addition, we have shown that LAS results in a (stochastically) worse delay distribution than any SMART policy in the many sources regime, and that the magnitude of this difference is highly dependent on the variability the job size distribution: as the variability increases, the difference in the decay rates of SMART and LAS increases. Further, the decay rates we derive for the SMART class and LAS provide insight into the “dominant” event of the delay distribution.

The fact that our results are derived under the many sources framework is a departure from much of the prior work studying the distribution of $W(k)$, which has used the large buffer framework. The difference is significant because the many sources framework allows the study of $\Pr(W(k) > m)$ for all m under a large number for flows, while the large buffer framework assumes a single flow and takes m to be very large.

From a methodological point of view, a key contribution of this paper is a two dimensional queue representation that enables: *(i)* the study of policies that depend on the job state (age and/or remaining size), as opposed to the queue length; and *(ii)* the study of a *class of policies*, as opposed to the analysis of individual policies. The second of these points is extremely important since practitioners can never implement the idealized policies, such as SRPT, that are typically the focus of theoretical research. However, by analyzing classes of policies defined by scheduling heuristics, theoretical results can hopefully be applied to the complex hybrid policies that are actually implemented. We believe that the two dimensional queue representation we introduce for analysis in the many sources large deviations framework provides a promising approach for the analysis of other priority-based scheduling policies and classes of policies in a large-scale regime – a regime of increasing relevance to modern computer systems such as routers and web servers.

9. REFERENCES

[1] M. F. Arlitt and C. L. Williamson. Internet web servers: Workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, 5(5):631–645, 1997.

[2] D. Bertsimas, I. C. Paschalidis, and J. N. Tsitsiklis. Asymptotic buffer overflow probabilities in multiclass multiplexers: An optimal control approach. *IEEE Trans. on Auto. Control*, 43:315–335, 1998.

[3] D. Botvich and N. Duffield. Large deviations, economies of scale, and the shape of the loss curve in large multiplexers. *Queueing Systems*, 20:293–320, 1995.

[4] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.

[5] M. E. Crovella, M. S. Taqqu, and A. Bestavros. *Heavy-tailed probability distributions in the World Wide Web: A Practical Guide To Heavy Tails*. Chapman and Hall, New York, 1998.

[6] G. de Veciana and J. Walrand. Effective bandwidths: Call admission, traffic policing and filtering for ATM networks. *Queueing Systems Theory and Applications*, 20:37–59, 1995.

[7] S. Delas, R. Mazumdar, and C. Rosenberg. Cell loss asymptotics for buffers handling a large number of independent stationary sources. In *Proc of IEEE Infocom*, volume 2, pages 551–558, 1999.

[8] S. Delas, R. Mazumdar, and C. Rosenberg. Tail asymptotics for HOL priority queues handling a large number of independent stationary sources. *Queue. Sys. Thry. and App.*, 40(2):183–204, 2002.

[9] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal. Implementation of SRPT scheduling in web servers. *ACM Trans. on Comp. Sys.*, 21(2), May 2003.

[10] C. Kotopoulos, N. Likhanov, and R. Mazumdar. Overflow asymptotics in GPS systems with heterogeneous longtailed inputs. In *Proc. of IEEE Infocom*, 2001.

[11] N. Likhanov and R. Mazumdar. Cell loss asymptotics for buffers fed with a large number of independent stationary sources. *Journal of Applied Probability*, 36:86–96, 1999.

[12] M. Mandjes and M. Nuyens. Sojourn time in the M/G/1 FB queue with light-tailed service times. *Prob. in the Eng. and Info. Sci.*, 19:351–361, 2005.

[13] L. Massoulie. Large deviations estimates for polling and weighted fair queueing service systems. *Adv. Perf. Anal.*, 2(2):103–128, 1999.

[14] D. McWherter, B. Schroeder, N. Ailamaki, and M. Harchol-Balter. Improving preemptive prioritization via statistical characterization of OLTP locking. In *Int. Conf on Data Engineering*, 2005.

[15] M. Nuyens, A. Wierman, and B. Zwart. Preventing large sojourn times using SMART scheduling. *Under Submission*, 2005.

[16] M. Nuyens and B. Zwart. A large-deviation analysis of GI/GI/1 SRPT queue. *Under Submission*, 2005.

[17] I. Paschalidis. Class-specific quality of service guarantees in multimedia communication networks. *Automatica, Special Issue on Control Methods for Communication Networks, V. Anantharam and J. Walrand, editors*, 35(12):1951–1969, 1999.

[18] I. Rai, G. Urvoy-Keller, and E. Biersack. Analysis of LAS scheduling for job size distributions with high variance. In *Proc. of ACM Sigmetrics*, 2003.

[19] I. Rai, G. Urvoy-Keller, M. Vernon, and E. Biersack. Performance modeling of LAS based scheduling in packet switched networks. In *Proc. of ACM Sigmetrics/Performance*, 2004.

[20] M. Rawat and A. Kshemkalyani. SWIFT: scheduling in web servers for fast response time. In *Symp. on Net. Comp. and App.*, 2003.

[21] R. Righter and J. Shanthikumar. Scheduling multiclass single server queueing systems to stochastically maximize the number of successful departures. *Prob. in the Eng. and Info. Sci.*, 3:967–978, 1989.

[22] R. Righter, J. Shanthikumar, and G. Yamazaki. On external service disciplines in single stage queueing systems. *Journal of Applied Probability*, 27:409–416, 1990.

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

[24] S. Shakkottai and R. Srikant. Many-sources delay asymptotics with applications to priority queues. *Queueing Systems: Theory and Applications*, 39:183–200, October 2001.

[25] A. Wierman and M. Harchol-Balter. Classifying scheduling policies with respect to unfairness in an M/GI/1. In *Proc. of ACM Sigmetrics*, 2003.

[26] A. Wierman, M. Harchol-Balter, and T. Osogami. Nearly insensitive bounds on SMART scheduling. In *Proc. of ACM Sigmetrics*, 2005.

[27] C. W. Yang and S. Shakkottai. Delay asymptote of the SRPT scheduler. In *Proceedings of the IEEE Conference on Decision and Control*, December 2004.