Queueing Syst
DOI 10.1007/s11134-015-9440-3

Instability of sharing systems in the presence of retransmissions

Predrag R. Jelenković[1] · **Evangelia D. Skiani**[1]

Received: 15 September 2014 / Revised: 30 January 2015

Abstract Retransmissions represent a primary failure recovery mechanism on all layers of communication network architecture. Similarly, fair sharing, for example, processor sharing (PS), is a widely accepted approach to resource allocation among multiple users. Recent work has shown that retransmissions in failure-prone, for example, wireless ad hoc, networks can cause heavy tails and long delays. In this paper, we discover a new phenomenon showing that PS-based scheduling induces complete instability with zero throughput in the presence of retransmissions, regardless of how low the traffic load may be. This phenomenon occurs even when the job sizes are bounded/fragmented, for example, deterministic. Our analytical results are further validated via simulation experiments. Moreover, our work demonstrates that scheduling one job at a time, such as first-come-first-serve, achieves a larger stability region and should be preferred in these systems.

Keywords Retransmissions · Restarts · Fair sharing · Instabilities · Processor sharing · Discriminatory processor sharing · Non-preemptive scheduling · GI/G/1 queue · Job fragmentation

Mathematics Subject Classification 60K25 · 68M20 · 93E15 · 60F99 · 90B18

Preliminary version of this paper has appeared earlier in SIGMETRICS'14 [1].
This work was supported by NSF Grant Number 0915784.

✉ Evangelia D. Skiani
valia@ee.columbia.edu

Predrag R. Jelenković
predrag@ee.columbia.edu

[1] Department of Electrical Engineering, Columbia University, New York, NY 10027, USA

Published online: 19 April 2015

1 Introduction

High variability and frequent failures characterize the majority of large-scale systems, for example, infrastructure-less wireless networks, cloud/parallel computing systems, etc. The nature of these systems imposes the employment of failure recovery mechanisms to guarantee their good performance. One of the most straightforward and widely used recovery mechanisms is to simply restart all the interrupted jobs from the beginning after a failure occurs. In communication systems, restart mechanisms lie at the core of the network architecture where retransmissions are used on all protocol layers to guarantee data delivery in the presence of channel failures, for example, automatic repeat request (ARQ) protocol [2], contention-based ALOHA-type protocols in the medium access control (MAC) layer, end-to-end acknowledgements in the transport layer, HTTP downloading scheme in the application layer, and others.

Furthermore, sharing is a primary approach to fair scheduling and efficient management of the available resources. Fair allocation of the network resources among different users can be highly beneficial for increasing throughput and utilization. For instance, CDMA is a multiple access method used in communication networks, where several users can transmit information simultaneously over a single channel via sharing the available bandwidth. Another example is processor sharing (PS) scheduling [3] where the capacity is equally shared between multiple classes of customers. In *generalized* PS (GPS) [4], service allocation is done according to some fixed weights. The related *discriminatory* PS (DPS) [5–7] is used in computing to model the weighted round-robin (WRR) scheduling, while it is also used in communications as a flow level model of heterogeneous TCP connections. Similarly, fair queuing (FQ) is a scheduling algorithm where the link capacity is fairly shared among active network flows; in weighted fair queuing (WFQ), which is the discretized version of GPS, different scheduling priorities are assigned to each flow.

In general, PS-based scheduling disciplines have been widely used in computer and communication networks. Early investigations of PS queues were motivated by applications in multiuser computer systems [8]. The M/G/1 PS queue has been studied extensively in the literature [9]. In the case of the M/M/1 PS system, the conditional Laplace transform of the waiting time was derived in [8]. The importance of scheduling in the presence of heavy tails was first recognized in [10], and later, in [11], the M/G/1 PS queue was studied assuming subexponential job sizes; see also [11] for additional references.

In [12], it was proven that, although there are policies known to optimize the sojourn time tail under a large class of heavy-tailed job sizes (for example, PS and SRPT) and there are policies known to optimize the sojourn time tail in the case of light-tailed job sizes, for example, first-come-first-serve (FCFS), no policies are known to optimize the sojourn time tail across both light- and heavy-tailed job size distributions. Indeed, such policies must "learn" the job size distribution in order to optimize the sojourn time tail. In the heavy-tailed scenarios, any scheduling policy that assigns the server exclusively to a very large job, for example, FCFS, may induce long delays, in which case sharing guarantees better performance.

In this paper, we study the effects of sharing on the system performance when restarts are employed in the presence of failures. We revisit the well-studied M/G/1 PS queue with a new focus on server failures and restarts. We use the following generic model, which was first introduced in [13] in the application context of computing. The system dynamics are described as a process $\{A_n\}_{n\geq 1}$, where A_n correspond to the periods when the system is available. $\{A_n\}_{n\geq 1}$ is a sequence of i.i.d random variables, independent of the job sizes. In each period of time that the system is available, say A_n, we attempt to execute a job of random size B. If $A_n > B$, we say that the job is successfully completed; otherwise, we restart the job from the beginning in the following period A_{n+1} when the channel is available.

With regard to retransmissions, it was first recognized in [13–15,20] that restart mechanisms may result in heavy-tailed (power law) delays even if the job sizes and failure rates are light tailed. In [16], it was shown that the power law delays arise whenever the cumulative hazard functions of the data and failure distributions are proportional. In the practically important case of bounded data units, a uniform characterization of the entire body of the retransmission distribution was derived in [17,18], which allows for determining the optimal size of data units/fragments in order to alleviate the power law effect. Later, these results were extended to the case where the channel is highly correlated [19], i.e., switches between states with different characteristics, and was proved that the delays are insensitive to the channel correlations and are determined by the 'best' channel state.

In this paper, our main contributions are the following. First, we prove that the M/G/1 PS queue is always unstable, regardless of how light the load is and how small the job sizes may be, see Theorems 1 and 2 in Sect. 3. This is a new phenomenon, since contrary to the conventional belief, sharing the service even between very small deterministic jobs can render the system completely unstable when retransmissions/restarts are employed. This instability is strong, in the sense of system having zero throughput. The intuition is the following. If a large number of jobs arrive in a short period of time, then under the elongated service time distribution induced by sharing, coupled with retransmissions, the queue will keep accumulating jobs that will equally share the capacity, which further exacerbates the problem. Every time a failure occurs, the system resets and the service requirement for each job elongates as the queue size increases. The expected delay until the system clears becomes increasingly long and, consequently, the queue will continue to grow, leading to instability. This result also applies to the DPS queue, where the service is not shared equally but according to some fixed weights. Next, we remove the Poisson assumption and extend our results to general renewal arrivals in Sect. 4. This demonstrates that instability arises from the interplay between sharing and retransmission/restart mechanisms, rather than any specific characteristics of the arrival process and/or service distribution.

We would also like to emphasize that job fragmentation cannot stabilize the system regardless of how small the fragments are made, since Theorem 2 shows instability for any minimum job size $\beta > 0$. Similarly, the system cannot be stabilized by checkpointing regardless of how small the intervals between successive checkpoints are chosen. In our experimental results, we make an interesting observation on the system behavior before it saturates. There exists a transient period, during which

the queue appears as if it were stable. Although it may occasionally accumulate a substantial number of jobs, it returns to zero and starts afresh. However, there exists a time when the queue reaches a critical size after which the service rate of the jobs reduces so much that neither of them can depart. Hence, as the queue continues to increase in size, the system becomes unstable.

To contrast these results, in Sect. 3.2, we study the stability of a non-preemptive policy that serves one job at a time under more specific assumptions of Poisson failure rates. To this end, Theorem 3 shows that when jobs are bounded, serving one job at a time, for example, FCFS, always has a non-empty stability region, and thus performs better than PS.

In order to gain further insight into the system, we then focus on its transient behavior and study the properties of the completion time of a finite number of jobs with no future arrivals. Specifically, we compare two work-conserving policies: scheduling one job at a time, for example, FCFS, and PS. Overall, we discover that serving one job at a time exhibits uniformly better performance than PS; compare Theorems 7 and 8, respectively. Furthermore, under more technical assumptions, and for light-tailed job/failure distributions, we show that PS performs distinctly worse compared to the heavy-tailed ones, and that PS is always unstable.

From an engineering perspective, our results indicate that traditional approaches in existing systems may be inadequate in the presence of failures. This new phenomenon demonstrates the need to revisit existing techniques for large-scale failure-prone systems, where PS-based scheduling may perform poorly. For example, since PS is unstable even for deterministic jobs, packet fragmentation, which is widely used in communications, cannot alleviate instabilities. Indeed, fragmentation can only postpone the time when the instability occurs, but cannot eliminate the phenomenon; see Example 1 in Sect. 6. Therefore, serving one job at a time, for example, FCFS, is highly advisable in such systems; see Sect. 3.2.

The paper is organized as follows. In Sect. 2, we introduce the model along with the necessary definitions and notation. Next, in Sect. 3, we present our main results on the M/G/1 PS queue, which are further extended in Sect. 4 to general renewal arrivals. On the other hand, in Sect. 3.2 we study the stability of non-preemptive policies that serve one job at a time, for example, FCFS. Later, in Sect. 5, we analyze the transient behavior of the system under two different scheduling policies, i.e., serving one job at a time and PS. Lastly, Sect. 6 presents our simulation experiments that validate our main theoretical findings, while Sect. 7 concludes the paper.

2 Definitions and notation

First, we provide the necessary definitions and notation assuming that the jobs are served individually. Consider a generic job of random size B, $B > 0$ a.s., requesting service in a failure-prone system. Without loss of generality, we assume that the system is of unit capacity. The failure dynamic is described as a process $\{A_n\}_{n\geq 1}$ of i.i.d availability periods, where at the end of each period A_n, the system experiences

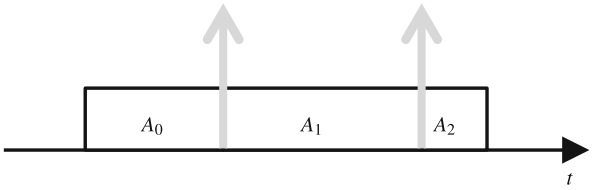


Fig. 1 System with failures

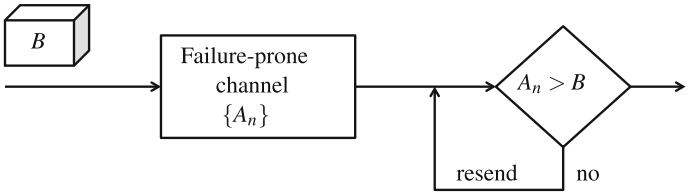


Fig. 2 Jobs executed in system with failures

a failure, as shown in Fig. 1. The channel/server statistics $\{A_n\}_{n\geq 1}$ are independent of the job size B.

Furthermore, we assume that the first failure occurs at time $A_0 \geq 0$, which is independent of $\{A_n\}_{n\geq 1}$ and B. When A_0 is equal in distribution to the excess/residual distribution of A_1, $\{A_n\}_{n\geq 0}$ will be in stationarity. Throughout the paper, we will use different assumptions on A_0, for example, $A_0 \equiv 0$, which will be explicitly stated in the corresponding results. Let A be a generic random variable that is equal in distribution to A_1. We denote the complementary cumulative distribution functions for A and B, respectively, as

$$\bar{G}(x) \triangleq \mathbb{P}(A > x) \qquad \text{and} \qquad \bar{F}(x) \triangleq \mathbb{P}(B > x).$$

At each period of time that the system becomes available, say A_n, we attempt to process a generic job of size B. If $A_n > B$, we say that the job is completed successfully; otherwise, we wait until the next period A_{n+1} when the channel is available and restart the job. A sketch of the model depicting the system is drawn in Fig. 2.

The number of restarts, N, and the total service time, S, for a job of size B, whose service begins immediately after a first failure A_0 and is served in isolation without preemption are defined as follows.

Definition 1 The number of restarts for a generic job of size B is defined as

$$N \triangleq \inf\{n \geq 1 : A_n > B\}.$$

Definition 2 The service time is the total time until a generic job of size B is successfully completed and is denoted by

$$S \triangleq \sum_{i=1}^{N-1} A_i + B.$$

Note that the preceding definitions will be different when jobs are sharing a server, as in the PS discipline. In general, a job B will successfully complete service during an availability period A_{n+1} if there exists $t \leq A_{n+1}$ such that

$$\int_{T_n}^{T_n+t} C_u^B \mathrm{d}u = B,$$

where C_u^B is the service rate that job B receives at time u and T_n is the time of the nth failure, $T_n = \sum_{i=0}^{n} A_i, n \geq 0$. Note that in general C_u^B depends on the number of jobs at time T_n, the arrival process, and the service discipline. We use B_j to denote the service requirement of job j where $\{B_j\}_{j\geq 1}$ is an i.i.d process equal in distribution to B. The failure times $\{A_n\}_{n\geq 0}$, job requirements $\{B_j\}_{j\geq 1}$ and the arrival process are mutually independent.

Throughout the paper, we use the following standard notation. For any two real functions $f(x)$ and $g(x)$ and fixed $x_0 \in \mathbb{R} \cup \{\infty\}$, we say $f(x) \sim g(x)$ as $x \to x_0$, to denote $\lim_{x\to x_0} f(x)/g(x) = 1$.

3 M/G/1 queue with restarts

In this section, we discuss the stability of the M/G/1 queue under two scheduling disciplines: PS and non-preemptive one job at a time policy. Throughout this section, we assume that the arrival process is Poisson with rate $\lambda > 0$. In the following subsection, we show in Theorem 2 that the M/G/1 PS queue is unstable under considerable generality. Next, in Sect. 3.2, we derive the necessary and sufficient condition for the system to be stable when the jobs are processed one at a time and the failure rates are Poisson.

3.1 Instability of processor sharing queue

In this section, we show in Theorems 1 and 2 that the M/G/1 PS queue is unstable when jobs need to restart after failures. We consider a general renewal failure process as defined in Sect. 2. First, in Proposition 1, we show that for some initial condition on the queue size, the probability that no job completes service approaches 1, under the mild assumption that jobs are bounded from below by some positive constant β. This is a natural assumption for communication or computing applications where jobs, for example, files, packets, threads, must have a header to contain the required information, such as destination address, thread id, etc. Hence, the job sizes, in practice, cannot be smaller than a positive constant.

Next, in Theorem 1, without any initial condition on the queue size, we prove that after some finite time, no job ever leaves the system; this result is stronger than standard stability theorems. Then, in Corollary 1, we draw a weaker conclusion that the queue size grows to infinity, which is also stated in Theorem 2. Nevertheless, the latter does not require the assumption on the minimum job size.

We begin with the following proposition. As previously mentioned, in this section we assume a general renewal failure process $\{A_n\}_{n\geq 0}$, as defined in Sect. 2. In the following proposition, we assume that the first failure occurs at $t = 0$, i.e., $A_0 = 0$.

The remaining results (Theorems 1 and 2) allow for an arbitrary delay until the first failure, $0 \leq A_0 < \infty$; this assumption includes the stationary version of $\{A_n\}_{n\geq 0}$, when A_0 has the excess distribution of A.

Proposition 1 *Assume that a failure occurs at time $t = 0$, i.e., $A_0 \equiv 0$, and there are $Q_0 \geq k$ jobs in the M/G/1 PS queue. If $\mathbb{E}A < \infty$ and $\mathbb{P}[B \geq \beta] = 1$, $\beta > 0$, then there exists $\theta > 0$, such that for all $k \geq 1$*

$$\mathbb{P}[\textit{no job ever completes service}] \geq 1 - O(\mathbb{E}A\mathbf{1}(A \geq \beta k) + e^{-\theta k}). \tag{3.1}$$

Proof Let $T_1 = \sum_{i=1}^{ck} A_i$ be the cumulative time that includes the first ck failures for $t > 0$; to simplify notation we write $\sum_x^y$ to denote $\sum_{\lceil x\rceil}^{\lfloor y\rfloor}$, where $\lceil x\rceil$ is the smallest integer $\geq x$ and $\lfloor y\rfloor$ is the largest integer $\leq y$. Now, define the event $\mathscr{A}_1 \equiv \mathscr{A}_1(k) \triangleq \{A_1 < \beta k, A_2 < \beta k, \ldots, A_{ck} < \beta k\}$. On this event, no job can leave the system since $Q_0 \geq k$ and all of them are at least of size β. Thus, if they were served in isolation, they could not have completed service in the first ck attempts.

Now, let E_1 denote the event that there is no departure in the first ck service attempts and there are at least k arrivals in $(0, T_1]$; we use $Z_{(t_0,t_1]}$ to denote the number of Poisson arrivals in the interval $(t_0, t_1]$, whereas we simply write Z_t for intervals $(0, t]$. Formally,

$$E_1 \supset \underline{E}_1 \triangleq \{Z_{T_1} \geq k, \mathscr{A}_1\},$$

on the set $\{Q_0 \geq k\}$. Note that $\underline{E}_1$ is clearly a subset of E_1, since there may be many other scenarios when no jobs leave the queue either because jobs are larger than β or more than k jobs are sharing the server. Now, observe that

$$\begin{aligned}\mathbb{P}(\underline{E}_1) &\geq \mathbb{P}(Z_{T_1} \geq k, T_1 \geq 2k/\lambda, \mathscr{A}_1)\\ &\geq \mathbb{P}(Z_{2k/\lambda} \geq k, T_1 \geq 2k/\lambda, \mathscr{A}_1)\\ &\geq \mathbb{P}(Z_{2k/\lambda} \geq k)\mathbb{P}(T_1 \geq 2k/\lambda, \mathscr{A}_1),\end{aligned}$$

since Poisson arrivals are independent of the failure process. Thus,

$$\mathbb{P}(\underline{E}_1) \geq \mathbb{P}(Z_{2k/\lambda} \geq k)\left(\mathbb{P}(\mathscr{A}_1) - \mathbb{P}(T_1 < 2k/\lambda)\right).$$

First, note that

$$\begin{aligned}\mathbb{P}(Z_{2k/\lambda} \geq k) &= 1 - \mathbb{P}(Z_{2k/\lambda} < k) = 1 - \mathbb{P}(2k - Z_{2k/\lambda} > k)\\ &\geq 1 - e^{-\theta k}\mathbb{E}e^{\theta(2k - Z_{2k/\lambda})} = 1 - e^{\theta k}\mathbb{E}e^{-\theta Z_{2k/\lambda}},\end{aligned}$$

by Cramer's bound for $\theta > 0$. Next, observe that $Z_{2k/\lambda}$ is Poisson with mean $2k$ and thus

$$\mathbb{P}(Z_{2k/\lambda} \geq k) \geq 1 - e^{\theta k}e^{2(e^{-\theta}-1)k} = 1 - e^{-\theta_1 k},$$

where $\theta_1 = 2(1 - e^{-\theta}) - \theta > 0$, for θ small.

Second, observe that

$$\mathbb{P}(T_1 < 2k/\lambda) = \mathbb{P}\left(\sum_{i=1}^{ck} A_i < 2k/\lambda\right) = \mathbb{P}\left(\sum_{i=1}^{ck} (A_i - \mathbb{E}A) < 2k/\lambda - ck\mathbb{E}A\right)$$
$$\leq \mathbb{P}\left(\sum_{i=1}^{3k/\lambda\mathbb{E}A} (\mathbb{E}A - A_i) > k/\lambda\right),$$

by picking $c \triangleq 3/(\lambda\mathbb{E}A)$. Now, let $X_i \triangleq \mathbb{E}A - A_i$, which are bounded from above since $X_i \leq \mathbb{E}A < \infty$, from our main assumption. Therefore, Cramer's large deviation bound implies that

$$\mathbb{P}(T_1 < 2k/\lambda) \leq \mathbb{P}\left(\sum_{i=1}^{3k/\lambda\mathbb{E}A} X_i > k/\lambda\right) \leq H_2 e^{-\theta_2 k},$$

for some $H_2, \theta_2 > 0$.

Therefore,

$$\mathbb{P}(\underline{E}_1) \geq (1 - e^{-\theta_1 k})\left(\mathbb{P}(\mathscr{A}_1) - H_2 e^{-\theta_2 k}\right)$$
$$\geq \mathbb{P}(A < \beta k)^{ck} - \left(e^{-\theta_1 k} + H_2 e^{-\theta_2 k} - H_2 e^{-(\theta_1+\theta_2)k}\right)$$
$$\geq (1 - \mathbb{P}(A \geq \beta k))^{ck} - He^{-\theta k},$$

where $\theta = \min(\theta_1, \theta_2)$ and $H > 0$ such that $H < (1 + H_2)$. Next, using $1 - x \geq e^{-2x}$ for small x, we have for all $k \geq k_0$

$$\mathbb{P}(\underline{E}_1) \geq e^{-2ck\mathbb{P}(A \geq \beta k)} - He^{-\theta k}$$
$$\geq 1 - 2ck\mathbb{P}(A \geq \beta k) - He^{-\theta k}$$
$$\geq e^{-4ck\mathbb{P}(A \geq \beta k) - 2He^{-\theta k}}.$$

Next, at time $\mathscr{T}_1 = T_1$, on event $\underline{E}_1$, the queue has at least $2k$ jobs, i.e., $Q_{\mathscr{T}_1} \geq 2k$, and no jobs have departed. Similarly, as before, let $T_2 = \sum_{i=ck+1}^{3ck} A_i$ be the cumulative time that includes the next $2ck$ failures, and define $\mathscr{A}_2 \equiv \mathscr{A}_2(k) = \{A_{ck+1} < 2\beta k, A_{ck+2} < 2\beta k, \ldots, A_{3ck} < 2\beta k\}$. Now, if E_2 is the event that there is no departure in the next $2ck$ attempts and there are at least $2k$ arrivals in $(\mathscr{T}_1, \mathscr{T}_2]$, then $E_2 \supset \underline{E}_2 \triangleq \{Z_{T_2} \geq 2k, \mathscr{A}_2\}$ on $\{Q_{\mathscr{T}_1} \geq 2k\}$; note that $\underline{E}_2$ is independent of $\underline{E}_1$. Then, the probability that no job departs in $(0, \mathscr{T}_2]$, where $\mathscr{T}_2 = T_1 + T_2$, is lower bounded by

$$\begin{aligned}\mathbb{P}(\text{no job departs in } (0, \mathscr{T}_2]) &\geq \mathbb{P}(E_1 \cap E_2)\\ &\geq \mathbb{P}(Z_{T_1} \geq k, \mathscr{A}_1, Z_{T_2} \geq 2k, \mathscr{A}_2) = \mathbb{P}(\underline{E}_1)\mathbb{P}(\underline{E}_2)\\ &\geq \mathbb{P}(Z_{T_1} \geq k, \mathscr{A}_1, Q_{\mathscr{T}_1} \geq 2k, Z_{(\mathscr{T}_1, \mathscr{T}_2]} \geq 2k, \mathscr{A}_2),\end{aligned} \tag{3.2}$$

since $\{Q_{\mathscr{T}_1} \geq 2k\} \supseteq \{Z_{T_1} \geq k, \mathscr{A}_1\}$ on the set $\{Q_0 \geq k\}$; the remaining statements in this proof should all be considered on $\{Q_0 \geq k\}$.

Next, via identical arguments to before, we obtain

$$\begin{aligned}\mathbb{P}(\underline{E}_2) &\geq \mathbb{P}(Z_{T_2} \geq 2k, T_2 \geq 4k/\lambda, \mathscr{A}_2)\\ &\geq \mathbb{P}(Z_{4k/\lambda} \geq 2k)\,(\mathbb{P}(\mathscr{A}_2) - \mathbb{P}(T_2 < 4k/\lambda)) \geq e^{-8ck\mathbb{P}(A \geq 2\beta k) - 2He^{-2\theta k}}.\end{aligned}$$

Therefore, at time $\mathscr{T}_2$, on event $\underline{E}_1 \cap \underline{E}_2$, there are at least $4k$ jobs.

In general, for any n, we can extend the reasoning from (3.2) to obtain

$$\begin{aligned}\mathbb{P}(\text{no job departs in } (0, \mathscr{T}_n]) &\geq \mathbb{P}(E_1 \cap E_2 \cap \cdots \cap E_n)\\ &\geq \mathbb{P}(Z_{T_1} \geq k, \mathscr{A}_1, Z_{T_2} \geq 2k, \mathscr{A}_2, \ldots, Z_{T_n} \geq 2^{n-1}k, \mathscr{A}_n)\\ &= \mathbb{P}(\underline{E}_1 \cap \underline{E}_2 \cap \cdots \cap \underline{E}_n),\end{aligned}$$

where $\mathscr{T}_n = \sum_{i=1}^{n} T_i$, $T_n = \sum_{i=(2^{n-1}-1)ck+1}^{(2^n-1)ck} A_i$, E_n is the event that there are no departures during $2^{n-1}ck$ attempts and there are at least $2^{n-1}k$ arrivals in $(\mathscr{T}_{n-1}, \mathscr{T}_n)$, and $\underline{E}_n = \{Z_{T_n} \geq 2^{n-1}k, \mathscr{A}_n\}$. Similarly to before,

$$\mathbb{P}(\underline{E}_n) \geq e^{-2^{n+1}ck\mathbb{P}(A \geq 2^{n-1}\beta k) - 2He^{-\theta 2^{n-1}k}}.$$

Hence, using the preceding inequality and the independence of $\underline{E}_i$s, we obtain

$$\begin{aligned}\mathbb{P}(E_1 \cap E_2 \cap \cdots \cap E_n) &\geq \mathbb{P}(\underline{E}_1 \cap \underline{E}_2 \cap \cdots \cap \underline{E}_n) = \mathbb{P}(\underline{E}_1)\mathbb{P}(\underline{E}_2)\cdots\mathbb{P}(\underline{E}_n)\\ &\geq \prod_{i=1}^{n} e^{-2^{i+1}ck\mathbb{P}(A \geq 2^{i-1}\beta k) - 2He^{-2^{i-1}\theta k}}\\ &= e^{-4\sum_{i=0}^{n-1} 2^i ck\mathbb{P}(A \geq 2^i\beta k) - 2H\sum_{i=0}^{n-1} e^{-2^i\theta k}}\\ &\geq e^{-4\sum_{i=0}^{\infty} 2^i ck\mathbb{P}(A \geq 2^i\beta k) - 2He^{-\theta k}\sum_{i=0}^{\infty} e^{-(2^i-1)\theta k}}.\end{aligned}$$

Now, observe that $\sum_{i=0}^{\infty} e^{-(2^i-1)\theta k} < \infty$, and thus we can pick H such that

$$\mathbb{P}(E_1 \cap E_2 \cap \cdots \cap E_n) \geq e^{-4\sum_{i=0}^{\infty} 2^i ck\mathbb{P}(A \geq 2^i\beta k) - He^{-\theta k}}.$$

Furthermore, we observe that

$$\begin{aligned}\sum_{i=0}^{\infty} 2^i ck\mathbb{P}(A \geq 2^i\beta k) &\leq \frac{c}{\beta}\sum_{i=0}^{\infty}\beta k\int_{2^i}^{2^{i+1}} \mathbb{P}(A \geq x\beta k)\mathrm{d}x \\ &\leq \frac{c}{\beta}\beta k\int_{1}^{\infty} \mathbb{P}(A \geq x\beta k)\mathrm{d}x = \frac{c}{\beta}\int_{\beta k}^{\infty} \mathbb{P}(A \geq y)\mathrm{d}y \\ &= \frac{c}{\beta}\mathbb{E}A\mathbf{1}(A \geq \beta k),\end{aligned}$$

and thus

$$\mathbb{P}(E_1 \cap E_2 \cap \cdots \cap E_n) \geq e^{-4c\beta^{-1}\mathbb{E}A\mathbf{1}(A\geq\beta k)-He^{-\theta k}} \geq 1 - H(\mathbb{E}A\mathbf{1}(A\geq\beta k)+e^{-\theta k}).$$

Lastly note that, on $\{Q_0 \geq k\}$,

$$\begin{aligned}\mathbb{P}(\text{no job ever completes service}) \geq \mathbb{P}(\cap_{i=1}^{\infty} E_i) &= \lim_{n\to\infty}\mathbb{P}(E_1 \cap E_2 \cap \cdots \cap E_n) \\ &\geq 1 - H(\mathbb{E}A\mathbf{1}(A \geq \beta k) + e^{-\theta k}),\end{aligned}$$

where the first inequality follows by definition and the second equality from monotone convergence.

Hence, we proved that the statement holds for all $k \geq k_0$. Lastly, for $k < k_0$, we can choose $H > 1/(\mathbb{E}A\mathbf{1}(A \geq \beta k_0) + e^{-\theta k_0})$, such that $\mathbb{P}$(no job ever completes service $|Q_0 \geq k) \geq 0 \geq 1 - H(\mathbb{E}A\mathbf{1}(A \geq \beta k_0) + e^{-\theta k_0}) \geq 1 - H(\mathbb{E}A\mathbf{1}(A \geq \beta k) + e^{-\theta k})$ and thus (3.1) holds trivially. □

We proceed with our main theorem which shows that, after some finite time, no job will ever depart.

Theorem 1 *In the M/G/1 PS queue, if* $\mathbb{E}A < \infty$, $0 \leq A_0 < \infty$ *a.s., and* $\mathbb{P}[B \geq \beta] = 1$, $\beta > 0$, *then*

$$\lim_{t\to\infty}\mathbb{P}\,(\textit{no job ever completes service after time } t) = 1.$$

Proof For any $k \geq 1$, let T_k be the first time that there are k jobs in the queue and a failure occurs. T_k is almost surely finite since it is upper bounded by the time $\bar{T}_k$ that there are at least k arrivals in an open interval of size β just before a failure; note that $0 \leq A_0 < \infty$ a.s. The probability of this event is $\mathbb{P}(Z_\beta \geq k) > 0$.

Let $\mathscr{B} \triangleq \{B_1^{T_k}, \ldots, B_{Q_{T_k}}^{T_k}\}$ denote the job sizes that are present in the queue at time T_k. From Proposition 1, we have

$$\mathbb{P}(\text{no job leaves after } T_k | Q_{T_k}, \mathscr{B}) \geq 1 - H(\mathbb{E}A\mathbf{1}(A \geq \beta k) + e^{-\theta k}) \geq 1 - \epsilon, \quad (3.3)$$

for all $k \geq k_0$, since $\theta > 0$ and $\mathbb{E}A\mathbf{1}(A \geq \beta k) \to 0$ as $k \to \infty$.

Now, for any fixed time t, we obtain

$$\begin{aligned}\mathbb{P}(\text{no job leaves after time } t) &\geq \mathbb{P}\left(T_k \leq t, \text{no job leaves after } T_k\right)\\ &= \mathbb{E}[\mathbb{P}(T_k \leq t | Q_{T_k}, \mathscr{B})\mathbb{P}(\text{no job leaves after } T_k | Q_{T_k}, \mathscr{B})]\\ &\geq \mathbb{P}(T_k \leq t)(1-\epsilon),\end{aligned}$$

which follows from (3.3); the equality follows from the fact that the event {no job leaves after T_k} is independent of the past, for example, $T_k \leq t$, given $Q_{T_k}, \mathscr{B}$. Next, recall that T_k is almost surely finite, i.e., $\lim_{t\to\infty} \mathbb{P}(T_k \leq t) = 1$, and thus taking the limit as $t \to \infty$ yields

$$\underline{\lim}_{t\to\infty} \mathbb{P}(\text{no job leaves after time} t) \geq 1-\epsilon.$$

Lastly, letting $\epsilon \downarrow 0$ finishes the proof. □

Corollary 1 *Under the conditions in Theorem 1, we have as* $t \uparrow \infty$,

$$Q_t \uparrow \infty \quad a.s.$$

Proof Note that the number of arrivals $Z_t \uparrow \infty$ as $t \uparrow \infty$ a.s. Thus, without loss of generality, we can assume that $Z_t(\omega) \uparrow \infty$ as $t \uparrow \infty$ for every ω (by excluding the set of zero probability). Then, for any $v > 0$,

$$U_v \triangleq \{\text{no job ever completes service after time } v\} \subset \{Q_t \uparrow \infty \text{ as } t \uparrow \infty\}.$$

Now, if $\omega \in U_v$, then for $t \geq v$, $Q_t(\omega)$ is non-decreasing. Furthermore, since there are no departures, the rate of increase of Q_t is equal to the arrival rate, and thus $Q_t \uparrow \infty$. Hence,

$$\mathbb{P}(Q_t \uparrow \infty \text{ as } t \uparrow \infty) \geq \mathbb{P}(\text{no job ever completes service after time } v)$$

which, by Theorem 1, implies

$$\mathbb{P}(Q_t \uparrow \infty \text{ as } t \uparrow \infty) = \lim_{v\to\infty} \mathbb{P}(\text{no job ever completes service after time } v) = 1.$$

□

Remark 1 Note that Theorem 1 is stronger than the standard stability theorems, since it also implies that eventually no job ever leaves the system.

Finally, we show instability, in general, without the condition $\mathbb{P}[B \geq \beta] = 1$. However, the conclusion is slightly weaker than in Theorem 1, and is the same as in Corollary 1. Basically, one cannot guarantee that no job ever completes service, since jobs can be arbitrarily small.

Theorem 2 *In the M/G/1 PS queue, if* $\mathbb{E}A < \infty$ *and* $0 \leq A_0 < \infty$ *a.s., we have as* $t \uparrow \infty$,

$$Q_t \uparrow \infty \quad a.s.$$

Proof First, by assumption, we can pick $\beta > 0$ such that $\mathbb{P}[B \geq \beta] > 0$. Then, for any time t, let Q_t^β be the number of jobs whose size is at least β and q_t^β be the number of jobs that are smaller than β. Hence,

$$Q_t = Q_t^\beta + q_t^\beta \geq \underline{Q}_t^\beta,$$

where $\underline{Q}_t^\beta$ is the queue in a system with the same arrival process where only jobs of size $B \geq \beta$ are served and the smaller ones are discarded. By Corollary 1, $\underline{Q}_t^\beta \uparrow \infty$ a.s., and, therefore, we obtain $Q_t \uparrow \infty$ a.s. □

3.1.1 Extension to DPS

In modern system design, PS cannot capture the heterogeneity of users and services, which is associated with unequal sharing of resources. Hence, we discuss the DPS queue which is a multi-class generalization of the PS queue: all jobs are served simultaneously at rates that are determined by a set of weights $w_i, i = 1, \ldots, K$. If there are n_j jobs in class j, each class-k job receives service at a rate $c_k = w_k / \sum_{j=1}^K w_j n_j$.

DPS has a broad range of applications. In computing, it is used to model WRR scheduling. In communication networks, DPS is used for modeling heterogeneous, for example, with different round trip delays, TCP connections. Despite the fact that the PS queue is well understood, the analysis of DPS has proven to be very hard; yet, our previous results on PS are easily extended to DPS in the corollary below.

Corollary 2 *Under the conditions in Theorems 1 and 2, the DPS queue is also always unstable, with the same conclusion as in Theorems 1 and 2, respectively.*

Proof Without loss of generality, assume that the set of weights is ordered such that $w_1 \leq w_2 \ldots \leq w_K$. In the M/G/1 DPS queue, the service allocation at any given time t for a single customer in class k is given by

$$c_k(t) = \frac{w_k}{\sum_{i=1}^K w_i n_i(t)} \leq \frac{w_k}{w_1 \sum_{i=1}^K n_i(t)} \leq \frac{w_K}{w_1 Q_t}.$$

Note that $c(t) = w_K/(w_1 Q_t)$ is the service rate in a PS queue with capacity $c = w_K/w_1 \geq 1$. Therefore, each class-k job, $k = 1 \ldots K$, in the DPS queue is served at a lower rate than the rate c of the PS queue. Hence,

$$Q_t^{DPS} \geq Q_t^{PS(c)},$$

and since, under the conditions in Theorem 1, the PS queue is always unstable, it follows that the DPS queue is also unstable. □

3.2 Stability of one job at a time non-preemptive policy

In this section, we study the stability of service disciplines where jobs are processed one at a time in a non-preemptive fashion, for example, FCFS. The stability results will be derived for exponentially distributed availability period A with rate μ. This assumption is needed to ensure the memoryless property of the system after each job completion.

Under such policies, the expected service time for a single job from Definition 2 is given by

$$\mathbb{E}[S] = \mathbb{E}\left[\sum_{i=1}^{N-1} A_i + B\right].$$

Note that $N \triangleq \inf\{n \geq 1 : A_n > B\}$ is a well-defined stopping time for the process $(A, \{A_n\}_{n\geq 1})$, and thus the expected service time follows from Wald's identity as

$$\begin{aligned}\mathbb{E}[S] &= \mathbb{E}\left[\sum_{i=1}^{N} A_i - A_N + B\right] \\ &= \mathbb{E}[N]\mathbb{E}[A] - \mathbb{E}[A_N] + \mathbb{E}[B].\end{aligned}$$

Now, assuming that the availability period A is exponentially distributed with rate μ (Poisson failures), the expected service time is given by

$$\begin{aligned}\mathbb{E}[S] &= \mathbb{E}[N]\mathbb{E}[A] - (\mathbb{E}[A] + \mathbb{E}[B]) + \mathbb{E}[B] \\ &= (\mathbb{E}[N] - 1)\mathbb{E}[A], \end{aligned} \tag{3.4}$$

since $\mathbb{E}[A_N] = \mathbb{E}\left[\mathbb{E}[A|A > B]\right] = \mathbb{E}[A+B] = \mathbb{E}[A]+\mathbb{E}[B]$, due to the memoryless property of the exponential distribution.

The necessary and sufficient condition for the stability of the non-preemptive M/G/1 queue with failures is

$$\lambda\mathbb{E}[S] < 1.$$

Next, we derive an explicit formula for $\mathbb{E}[N]$ by observing that

$$\mathbb{P}[N > n|B] = \mathbb{P}(A \leq B|B)^n = G(B)^n.$$

Thus, using the exponential distribution of A, the expected number of restarts is

$$\begin{aligned}\mathbb{E}[N] = \mathbb{E}[\mathbb{E}[N|B]] &= \mathbb{E}\left[\sum_{n=0}^{\infty} \mathbb{P}[N > n|B]\right] \\ &= \mathbb{E}\left[\sum_{n=0}^{\infty} G(B)^n\right] = \mathbb{E}\left[\bar{G}(B)^{-1}\right] = \mathbb{E}[e^{\mu B}].\end{aligned}$$

Hence, plugging the preceding expression in (3.4), we obtain

$$\mathbb{E}[S] = (\mathbb{E}[e^{\mu B}] - 1)\mu^{-1},$$

which yields the following theorem.

Theorem 3 *If $\{A_n\}_{n\geq 0}$ is Poisson with rate μ, arrivals are Poisson with rate $\lambda > 0$, and B is a typical job size, then the queue, for any non-preemptive policy that serves one job at a time, for example, FCFS, is stable iff*

$$\lambda \mathbb{E}[S] = \lambda \mu^{-1} \left(\mathbb{E}[e^{\mu B}] - 1 \right) < 1. \tag{3.5}$$

Note that, for exponential job sizes, the mean service time is finite and equal to $1/(1/\mathbb{E}[B] - \mu)$ if and only if $\mathbb{E}[B] < 1/\mu$, and the stability region is given by $\lambda/(1/\mathbb{E}[B] - \mu) < 1$. On the other hand, if B does not have exponential moments, then $\mathbb{E}[S] = \infty$, i.e., any non-preemptive policy will be unstable. Furthermore, the stability region for the system with failures is strictly smaller than in the traditional M/G/1 queue, since $\mathbb{E}[S] > \mu^{-1}\mu\mathbb{E}[B] = \mathbb{E}[B]$. In addition, since $e^x - 1 - x$ is increasing in x for $x > 0$, the stability region shrinks as the jobs grow in size. Alternatively, as the job sizes are decreasing, for example, applying fragmentation/checkpointing techniques, the stability region of a system with failures can approach the one of the traditional M/G/1 queue. Specifically, if $B = \beta$ is deterministic, $\lambda\mu^{-1}(e^{\mu\beta} - 1) \sim \lambda\beta$ as $\beta \to 0$, where $\lambda\beta < 1$ is the stability region of the ordinary M/G/1 queue without failures.

Remark 2 Note that the preceding result can be derived alternatively by noticing that for deterministic job sizes, $B = \beta$, the service time S behaves exactly the same as a busy period in an M/D/∞ queueing system with arrival rate μ and service time B, which yields $\mathbb{E}[S] = (e^{\mu\beta} - 1)/\mu$. This line of argument extends to random job sizes B, as in Theorem 3.

4 GI/G/1 PS queue with restarts

In the previous section, we showed that PS is unstable assuming Poisson arrivals. Here, we show that this result can be extended to more general arrival distributions, for example, renewal processes. However, to avoid technical complications we assume that the failure process is Poisson of rate μ, i.e., the availability periods A_i are exponential. To this end, we use $M_{(t_0,t_1]}$ to denote the number of Poisson failures in $(t_0, t_1]$ and write M_t for intervals of the form $(0, t]$. Let $(\tau, \{\tau_n\}_{n\geq 1})$ be an i.i.d. sequence, where τ_n represent the interarrival times of the renewal process. Similarly to the definition of the general failure process in Sect. 2, we assume that the first arrival occurs at time $\tau_0 \geq 0$. When τ_0 has the residual distribution of τ_1, then $\{\tau_n\}_{n\geq 0}$ will be in stationarity.

The main purpose of this section is to show that there is nothing special about the Poisson arrival assumption that leads to instability. Instead, the instability results from the interplay between sharing and retransmission/restart mechanisms. First, we prove the following proposition using similar arguments to those in Proposition 1. However, we embed the proof at the points of arrivals instead of failures. In the following proposition, we assume that the first arrival occurs at $t = 0$, i.e., $\tau_0 = 0$. The remaining results allow for an arbitrary delay until the first arrival, $0 \leq \tau_0 < \infty$;

these results imply the stationary version of $\{\tau_n\}_{n\geq 0}$, when τ_0 has the excess distribution of τ_1.

Proposition 2 *Assume that a new job arrives at time $t = 0$, i.e., $\tau_0 = 0$, and there are $Q_0 \geq k$ jobs in the GI/G/1 PS queue with remaining service $\geq \beta$. If failures are Poisson, $\mathbb{E}\tau^{1+\delta} < \infty, 0 < \delta < 1$ and $\mathbb{P}[B \geq \beta] = 1, \beta > 0$, then for all $k \geq 1$*

$$\mathbb{P}\left[\textit{no job ever completes service}\right] \geq 1 - O(\mathbb{E}A\mathbf{1}(A \geq \beta k) + k^{-\delta}).$$

Proof Let $T_1 = \sum_{i=1}^{k} \tau_i$ be the cumulative time that includes the first k arrivals for $t > 0$ and M_{T_1} be the number of failures in $(0, T_1)$. Now, define the event $\mathscr{A}_1 \equiv \mathscr{A}_1(k) \triangleq \{A_1 < \beta k, A_2 < \beta k, \ldots, A_{M_{T_1}} < \beta k\}$. On this event, no job can leave the system since $Q_0 \geq k$ and all of them are at least of size β. Thus, if they were served in isolation, they could not have completed service in the first M_{T_1} attempts.

Now, with a small abuse of notation, let E_1 denote the event that there is no departure in the first M_{T_1} attempts and there are at most ck failures in $(0, T_1]$. Formally,

$$E_1 \supset \underline{E}_1 \triangleq \{M_{T_1} \leq ck, \mathscr{A}_1\},$$

on the set $\{Q_0 \geq k\}$. Now, observe that

$$\begin{aligned}
\mathbb{P}(\underline{E}_1) &= \mathbb{P}(M_{T_1} \leq ck, A_1 < \beta k, A_2 < \beta k, \ldots, A_{M_{T_1}} < \beta k) \\
&\geq \mathbb{P}(M_{T_1} \leq ck, A_1 < \beta k, A_2 < \beta k, \ldots, A_{ck} < \beta k) \\
&\geq \mathbb{P}(A_1 < \beta k)^{ck} - \mathbb{P}(M_{T_1} > ck).
\end{aligned}$$

Next, note that

$$\begin{aligned}
\mathbb{P}(M_{T_1} > ck) &= \mathbb{P}\left(M_{T_1} > ck, T_1 \leq \frac{3k\mathbb{E}\tau}{2}\right) + \mathbb{P}\left(M_{T_1} > ck, T_1 > \frac{3k\mathbb{E}\tau}{2}\right) \\
&\leq \mathbb{P}\left(M_{\frac{3k\mathbb{E}\tau}{2}} > ck\right) + \mathbb{P}\left(T_1 > \frac{3k\mathbb{E}\tau}{2}\right),
\end{aligned}$$

where the first term is negligible for $c > 2\mu\mathbb{E}\tau$ since the expected number of failures is $3k\mu\mathbb{E}\tau/2$. Now, observe that

$$\mathbb{P}\left(T_1 > \frac{3k\mathbb{E}\tau}{2}\right) = \mathbb{P}\left(\sum_{i=1}^{k} \tau_i > \frac{3k\mathbb{E}\tau}{2}\right) = \mathbb{P}\left(\sum_{i=1}^{k} (\tau_i - \mathbb{E}\tau) > \frac{3k\mathbb{E}\tau}{2} - k\mathbb{E}\tau\right).$$

Now, let $X_i \triangleq \tau_i - \mathbb{E}\tau$, and by choosing $h = 2^{-\delta}(\mathbb{E}\tau)^{1+\delta}$ and $y = \mathbb{E}\tau/4$ in Lemma 1 of [20], we obtain

$$\mathbb{P}\left(\sum_{i=1}^{k} X_i > k\mathbb{E}\tau/2\right) \le k\mathbb{P}(X_1 > k\mathbb{E}\tau/4) + \frac{hk}{2^{-\delta}(k\mathbb{E}\tau)^{1+\delta}}$$
$$\le k\mathbb{P}(\tau_1 > k\mathbb{E}\tau/4 + \mathbb{E}\tau) + \frac{1}{k^{\delta}}$$
$$\le k\frac{\mathbb{E}\tau^{1+\delta}}{(k\mathbb{E}\tau/4 + \mathbb{E}\tau)^{1+\delta}} + k^{-\delta} \le 2k^{-\delta}.$$

Therefore,

$$\mathbb{P}(\underline{E}_1) \ge (1 - \mathbb{P}(A \ge \beta k))^{ck} - 2k^{-\delta},$$

where using $1 - x \ge e^{-2x}$ for small x, we have for all $k \ge k_0$

$$\mathbb{P}(\underline{E}_1) \ge e^{-2ck\mathbb{P}(A\ge\beta k)} - 2k^{-\delta} \ge 1 - 2ck\mathbb{P}(A \ge \beta k) - 2k^{-\delta}$$
$$\ge e^{-4ck\mathbb{P}(A\ge\beta k) - 4k^{-\delta}}.$$

Next, at time $\mathscr{T}_1 = T_1$, on event $\underline{E}_1$, the queue has at least $2k$ jobs, i.e., $Q_{\mathscr{T}_1} \ge 2k$, and no jobs have departed. Similarly to before, let $T_2 = \sum_{i=k}^{3k} \tau_i$ be the cumulative time that includes the next $2k$ arrivals, and define $\mathscr{A}_2 \equiv \mathscr{A}_2(k) = \{A_{M_{T_1}+1} < 2\beta k, A_{M_{T_1}+2} < 2\beta k, \ldots, A_{M_{T_1}+T_2} < 2\beta k\}$. The probability that no job departs in $(0, \mathscr{T}_2]$, where $\mathscr{T}_2 = T_1 + T_2$, is lower bounded by

$$\mathbb{P}(\text{no job departs in } (0, \mathscr{T}_2]) \ge \mathbb{P}(M_{T_1} \le ck, \mathscr{A}_1, Q_{\mathscr{T}_1} \ge 2k, M_{(\mathscr{T}_1, \mathscr{T}_2]} \le 2ck, \mathscr{A}_2)$$
$$\ge \mathbb{P}(M_{T_1} \le ck, \mathscr{A}_1, M_{(\mathscr{T}_1, \mathscr{T}_2]} \le 2ck, \mathscr{A}_2), \tag{4.1}$$

since $\{Q_{\mathscr{T}_1} \ge 2k\} \supseteq \{M_{T_1} \le ck, \mathscr{A}_1\}$ on the set $\{Q_0 \ge k\}$; to avoid repetitions, the following statements are all on $Q_0 \ge k$.

Now, if E_2 is the event that there is no departure in the next M_{T_2} attempts and there are at most $2ck$ failures in $(\mathscr{T}_1, \mathscr{T}_2]$, then $E_2 \supset \underline{E}_2 \triangleq \{M_{T_2} \le 2ck, \mathscr{A}_2\}$; note that $\underline{E}_2$ is independent of $\underline{E}_1$ due to the Poisson memoryless property. Via identical arguments to before, we obtain

$$\mathbb{P}(\underline{E}_2) \ge \mathbb{P}(M_{T_2} \le 2ck, A_{ck+1} < \beta k, \ldots, A_{3ck} < \beta k)$$
$$\ge e^{-8ck\mathbb{P}(A\ge 2\beta k) - 4(2k)^{-\delta}}.$$

Therefore, at time $\mathscr{T}_2$, on event $\underline{E}_1 \cap \underline{E}_2$, there are at least $4k$ jobs.

In general, for any n, we can extend the reasoning from (4.1) to obtain

$$\mathbb{P}(\text{no job departs in } (0, \mathscr{T}_n]) \ge \mathbb{P}(M_{T_1} \le ck, \mathscr{A}_1, M_{T_2} \le 2ck,$$
$$\mathscr{A}_2, \ldots, M_{T_n} \le 2^{n-1}k, \mathscr{A}_n)$$
$$= \mathbb{P}(\underline{E}_1 \cap \underline{E}_2 \cap \cdots \cap \underline{E}_n),$$

where $\mathscr{T}_n = \sum_{i=1}^{n} T_i$, $T_n = \sum_{i=(2^{n-1}-1)k+1}^{(2^n-1)k} \tau_i$, E_n denotes the event that there is no departure in M_{T_n} attempts and there are at most 2^{n-1} failures in $(\mathscr{T}_{n-1}, \mathscr{T}_n]$, and $\underline{E}_n = \{M_{T_n} \le 2^{n-1}ck, \mathscr{A}_n\}$. Similarly,

$$\mathbb{P}(\underline{E}_n) \geq e^{-2^{n+1}ck\mathbb{P}(A\geq 2^{n-1}\beta k)-4(2^{n-1}k)^{-\delta}}.$$

Hence, we obtain

$$\begin{aligned}\mathbb{P}(E_1 \cap E_2 \cap \cdots \cap E_n) &\geq \prod_{i=1}^{n} e^{-2^{i+1}ck\mathbb{P}(A\geq 2^{i-1}\beta k)-4(2^{i-1}k)^{-\delta}} \\ &= e^{-4\sum_{i=0}^{n-1} 2^i ck\mathbb{P}(A\geq 2^i\beta k)-4k^{-\delta}\sum_{i=0}^{n-1}(2^i)^{-\delta}} \\ &\geq e^{-4\sum_{i=0}^{\infty} 2^i ck\mathbb{P}(A\geq 2^i\beta k)-4k^{-\delta}\sum_{i=0}^{\infty}2^{-\delta i}}.\end{aligned}$$

Now, observe that $\sum_{i=0}^{\infty} 2^{-\delta i} < \infty$, and thus we can pick $H > 0$ such that

$$\mathbb{P}(E_1 \cap E_2 \cap \cdots \cap E_n) \geq e^{-4\sum_{i=0}^{\infty} 2^i ck\mathbb{P}(A\geq 2^i\beta k)-Hk^{-\delta}}.$$

The remainder of the proof follows identical arguments to those in Proposition 1. Thus, on $\{Q_0 \geq k\}$,

$$\mathbb{P}(\text{no job ever completes service}) \geq 1 - H(\mathbb{E}A\mathbf{1}(A \geq \beta k) + k^{-\delta}).$$

□

Theorem 4 *In the GI/G/1 PS queue, if failures are Poisson,* $0 \leq \tau_0 < \infty$ *a.s.,* $\mathbb{E}\tau^{1+\delta} < \infty, 0 < \delta < 1$ *and* $\mathbb{P}[B \geq \beta] = 1, \beta > 0$*, then*

$$\lim_{t\to\infty} \mathbb{P}\,(\textit{no job ever completes service after time } t) = 1.$$

Proof Similarly to the proof of Theorem 1, we observe the system at time V_k when there are k jobs in the queue and a failure occurs. Since the arrivals are non-Poisson, we need additional reasoning to ensure that $V_k < \infty$ a.s. In this regard, let us consider a time interval $T_1 = \sum_{i=1}^{k} \tau_i$ when the first k arrivals occur. Then, let t_k be such that $\mathbb{P}(T_1 < t_k) > 0$ and divide t_k into smaller intervals of size β. Now, consider the probability that $\{T_1 < t_k\}$ and there is at least one failure in each of the small intervals of size β. Since the failures are Poisson, this event has a positive, albeit extremely small, probability. If this event occurs, then $V_k \leq T_1 < \infty$ a.s. Otherwise, repeat the procedure on the next interval $T_2 = \sum_{i=k+1}^{2k} \tau_i$. Since the arrivals are renewal and failures are Poisson, the desired event in interval T_2 is independent and has the same probability as in T_1. Hence, after a geometric number of attempts, the queue will have at least k jobs at the time of failure, implying that $V_k < \infty$ a.s.

Now, the remainder of the proof follows the same arguments as in Theorem 1 of Sect. 3. We omit the details. □

Similarly to Theorem 2 of Sect. 3, we drop the condition $\mathbb{P}[B \geq \beta] = 1$ and prove general instability.

Theorem 5 *In the GI/G/1 PS queue, if failures are Poisson,* $0 \leq \tau_0 < \infty$ *a.s., and* $\mathbb{E}\tau^{1+\delta} < \infty, 0 < \delta < 1$*, we have as* $t \uparrow \infty$*,*

$$Q_t \uparrow \infty \quad a.s.$$

The proof is similar to the proof of Theorem 2 and thus is omitted. Furthermore, the equivalent results could be stated for the DPS scheduler as well. Lastly, the preceding findings could be further extended to both non-Poisson arrivals and non-Poisson failures. However, the proofs would be much more involved and complicated; here, we avoid such technicalities.

5 Transient behavior: scheduling a finite number of jobs

In the previous sections, we focused on the steady-state behavior of the M/G/1 queue with restarts and proved that PS is always unstable for failure distributions with finite first moment. We also showed instability for the GI/G/1 PS queue, assuming Poisson failures. In this section, in order to gain further insight into this system, we study its transient behavior. In this regard, we consider a queue with a finite number of jobs and no future arrivals and compute the total time until all jobs are completed. In Sects. 5.1 and 5.2, we analyze the system performance when the jobs are served one at a time and when PS is used, respectively. More precisely, for a finite number of jobs with sizes B_i, $1 \leq i \leq m$, and assuming no future arrivals, we study the completion time Θ_m, until all m jobs complete their service. Throughout this section, we assume that service starts at $t = 0$ and $A_0 \equiv 0$; furthermore, we assume that the distribution functions $\bar{G}(x)$ and $\bar{F}(x)$ are absolutely continuous for all $x \geq 0$.

Note that in the case of traditional work-conserving scheduling systems the completion time does not depend on the scheduling discipline and is always simply equal to $\sum_{i=1}^{m} B_i$. However, in channels with failures there can be a stark difference in the total completion time depending on the scheduling policy. This difference can be so large that in some systems the expected completion time can be infinite, while in others finite, or even having many high moments.

Overall, we discover that, with respect to the distribution of the total completion time Θ_m, serving one job at a time exhibits uniformly better performance than PS; see Theorems 7 and 8. Furthermore, when the cumulative hazard functions of the job and failure distributions are proportional, i.e., $\log \bar{F}(x) \sim \alpha \log \bar{G}(x)$, we show that PS performs distinctly worse for the light-tailed job/failure distributions as opposed to the heavy-tailed ones; see parts (i) and (ii) of Theorem 8.

Before presenting our main results, we state the following theorem on the logarithmic asymptotics of the time $\bar{S} = \sum_{i=1}^{N} A_i = S + (A_N - B)$, where S is from Definition 2. Note that $\bar{S}$ includes the remaining time $(A_N - B)$ until the next channel availability period, thus representing a natural upper bound for S. In the following, let $\vee \equiv \max$.

Theorem 6 *If* $\log \bar{F}(x) \sim \alpha \log \bar{G}(x)$ *as* $x \to \infty$, $\alpha > 1$, $\mathbb{E}[B^{\alpha+\delta}] < \infty$, *and* $\mathbb{E}[A^{1 \vee \alpha}] < \infty$ *for some* $\delta > 0$, *then*

$$\lim_{t \to \infty} \frac{\log \mathbb{P}[\bar{S} > t]}{\log t} = -\alpha \tag{5.1}$$

Proof By Theorem 6 in [20], when specialized to the conditions of this theorem, we obtain that $\log \mathbb{P}[S > t] \to -\alpha \log t$ as $t \to \infty$. This immediately yields the lower

bound for $\bar{S} = S + (A_N - B) \geq S$. For the upper bound, $\bar{S} = S + (A_N - B)$ and the union bound result in

$$\mathbb{P}[\bar{S} > 2x] \leq \mathbb{P}[S > x] + \mathbb{P}[A_N - B > x].$$

Hence, in view of Theorem 6 in [20], we only need to bound $\mathbb{P}[A_N - B > x]$. To this end, observe that

$$\begin{aligned}
\mathbb{P}[A_N - B > x] &= \mathbb{P}[A_N > B + x] = \sum_{i=1}^{\infty} \mathbb{P}[A_i > B + x, N = i] \\
&= \sum_{i=1}^{\infty} \mathbb{P}\left[A_i > B + x, A_1 < B, \ldots, A_{i-1} < B\right] \\
&= \sum_{i=1}^{\infty} \mathbb{E}\left[\mathbb{P}(A_i > B + x | B)\, \mathbb{P}(A_1 < B | B)^{i-1}\right] \\
&= \mathbb{E}\left[\frac{\bar{G}(B + x)}{\bar{G}(B)}\right] \leq \bar{G}(x)\mathbb{E}[N],
\end{aligned}$$

since $\mathbb{E}[N] = \mathbb{E}(1/\bar{G}(B))$. Now, the condition $\alpha > 1$ guarantees that $\mathbb{E}[N] < \infty$, whereas $\mathbb{E}[A^{\alpha}] < \infty$ implies that $\bar{G}(x) = O(1/x^{\alpha})$. Thus, (5.1) is satisfied. □

5.1 One job at a time non-preemptive policy

In this subsection, we consider the failure-prone system that was introduced in Sect. 2, with unit capacity. The jobs are served one at a time, for example, FCFS. Herein, we analyze the performance of this system assuming that, initially, there are m jobs in the queue and there are no future arrivals. Specifically, we study the total completion time, which is defined below.

Definition 3 The total completion time is defined as the total time until all the jobs are successfully completed and is denoted by

$$\Theta_m \triangleq \sum_{i=1}^{m} S_i,$$

where m is the total number of jobs in the system and S_i is the service requirement for each job.

In the following theorem, we prove that the tail asymptotics of the total completion time, from Definition 3, under this policy is a power law of the same index as the service time of a single job.

Theorem 7 *If* $\log \bar{F}(x) \sim \alpha \log \bar{G}(x)$ *as* $x \to \infty$, $\alpha > 1$, $A_0 = 0$, $\mathbb{E}[B^{\alpha+\delta}] < \infty$, *and* $\mathbb{E}[A^{1 \vee \alpha}] < \infty$ *for some* $\delta > 0$, *then*

$$\lim_{t \to \infty} \frac{\log \mathbb{P}[\Theta_m > t]}{\log t} = -\alpha.$$

Proof Recall that the service requirement for a job B_i was previously defined as $S_i = \sum_{j=1}^{N_i-1} A_j + B_i$.

For the *lower* bound, we observe that

$$\mathbb{P}[\Theta_m > t] \geq \mathbb{P}[S_1 > t],$$

since the total completion time is at least equal to the service time of a single job. By taking the logarithm and using Theorem 6 in [20], we have

$$\frac{\log \mathbb{P}[\Theta_m > t]}{\log t} \geq -(1+\epsilon)\alpha. \tag{5.2}$$

For the *upper* bound, we compare Θ_m with the completion time in a system where the server is kept idle between the completion time of the previous job and the next failure. Clearly,

$$\Theta_m \leq \bar{\Theta}_m \triangleq \sum_{i=1}^{m} \bar{S}_i,$$

where $\bar{S}_i \triangleq \sum_{j=1}^{N_i} A_j$ are the service times that include the remaining availability period A_{N_i}.

Then, we argue that

$$\mathbb{P}[\Theta_m > t] \leq \mathbb{P}\left[\sum_{i=1}^{m} \bar{S}_i > t\right] \leq m\mathbb{P}\left[\bar{S}_1 > \frac{t}{m}\right],$$

which follows from the union bound. By taking the logarithm and using Theorem 6, we have

$$\frac{\log \mathbb{P}[\Theta_m > t]}{\log t} \leq -\alpha(1-\epsilon) + \frac{\log m}{\log t} \leq -(1-2\epsilon)\alpha, \tag{5.3}$$

where we pick t large enough such that $\log t \geq \log m/(\alpha\epsilon)$.

Letting $\epsilon \to 0$ in both (5.2) and (5.3) finishes the proof. □

5.2 Processor sharing discipline

In this subsection, we analyze the PS discipline where m jobs share the (unit) capacity of a single server. We present our main theorem on the logarithmic scale, which shows that the tail asymptotics of the total completion time are determined by the shortest job in the queue. In particular, under our main assumptions, this time is a power law, but it exhibits a different exponent depending on the job size distribution, as our results demonstrate; see Theorem 8 and the proof.

– If the jobs are subexponential (heavy tailed) or exponential, the total delay is simply determined by the time required for any single job to complete its service, as if it were the only one present in the queue.

– If the jobs are superexponential (light tailed), the total delay is determined by the service time of the *shortest* job. This job generates the heaviest asymptotics among all the rest.

Our main result, stated in Theorem 8 below, shows that on the logarithmic scale the distribution of the total completion time Θ_m^{PS} is heavier by a factor $m^{\gamma-1}$ for superexponential jobs relative to the subexponential or exponential case, when the cumulative hazard functions F and G are proportional. Therefore, in systems with failures and restarts, sharing the capacity among light-tailed jobs induces long delays, whereas, for heavy-tailed ones, PS appears to perform as well as serving the jobs one at a time. Interestingly enough, this deterioration in performance is determined by the time it takes to serve the shortest job in the system.

Note that in a PS queue with no future arrivals, the shortest job will depart first. Immediately after this, the server will continue serving the remaining $m-1$ jobs, and, similarly, the shortest job, i.e., the second shortest among the original m jobs, will depart before all the others. This pattern will continue until the departure of the largest job, which is served alone.

Theorem 8 *Assume that the cumulative hazard function* $-\log \bar{F}(x)$ *is regularly varying with index* $\gamma \geq 0$. *If* $\log \bar{F}(x) \sim \alpha \log \bar{G}(x)$ *as* $x \to \infty$, $\alpha > 1$, $A_0 = 0$, $\mathbb{E}[B^{\alpha+\delta}] < \infty$, *and* $\mathbb{E}[A^{1\vee\alpha}] < \infty$ *for some* $\delta > 0$, *then*

(*i*) *if* $\gamma \leq 1$, *i.e., B is subexponential or exponential, then*

$$\lim_{t\to\infty} \frac{-\log \mathbb{P}[\Theta_m^{PS} > t]}{\log t} = \alpha,$$

(*ii*) *if* $\gamma > 1$, *i.e., B is superexponential, then*

$$\lim_{t\to\infty} \frac{-\log \mathbb{P}[\Theta_m^{PS} > t]}{\log t} = \frac{\alpha}{m^{\gamma-1}} < \alpha.$$

Remark 3 When $\alpha > 1$, we easily verify that $\mathbb{E}[\Theta_m^{PS}] < \infty$ in case (*i*); if the jobs are superexponential, for example, case (*ii*), then $\mathbb{E}[\Theta_m^{PS}] = \infty$ if $\alpha < m^{\gamma-1}$.

Proof Let $B^{(1)} \leq B^{(2)} \leq \cdots \leq B^{(m)}$ be the order statistics of the jobs $B_1, B_2, \ldots, B_m$.

The assumption that $-\log \bar{F}(x)$ is regularly varying with index γ implies that

$$\log \bar{F}(\lambda x) \sim \lambda^{\gamma} \log \bar{F}(x), \tag{5.4}$$

for any $\lambda > 0$.

We begin with the *lower* bound.

(*i*) *Subexponential or exponential jobs* ($\gamma \leq 1$). The total completion time is lower bounded by the time required for a single job to depart when it is exclusively served, i.e., if the total capacity of the system is used. Hence, it follows that

$$\mathbb{P}[\Theta_m^{PS} > t] \geq \mathbb{P}[S_1 > t], \tag{5.5}$$

where S_1 is the service time of a single job of random size B_1, when there are no other jobs in the system. Now, recalling Theorem 6 in [20], it holds that

$$\lim_{t\to\infty} \frac{\log \mathbb{P}[S_1 > t]}{\log t} = -\alpha.$$

By taking the logarithm in (5.5), the lower bound follows immediately.

(*i*) *Superexponential jobs* ($\gamma > 1$).

The total completion time is lower bounded by the delay experienced by the shortest job, and hence,

$$\mathbb{P}[\Theta_m^{PS} > t] \geq \mathbb{P}[S_1^{PS} > t],$$

where S_1^{PS} is the service time of job $B^{(1)}$. First, note that the distribution of $B^{(1)}$ is given by

$$\begin{aligned}\mathbb{P}(B^{(1)} > x) &= \mathbb{P}(B_1 > x, B_2 > x, \ldots, B_m > x)\\ &= \mathbb{P}(B_1 > x)\mathbb{P}(B_2 > x)\cdots\mathbb{P}(B_m > x)\\ &= \mathbb{P}(B_1 > x)^m = \bar{F}(x)^m,\end{aligned} \tag{5.6}$$

since $B_i, i = 1, \ldots, m$, are independent and identically distributed. Now, using (5.6) and (5.4), together with our main assumption, we observe that

$$\begin{aligned}\log \mathbb{P}(mB^{(1)} > x) &= m \log \bar{F}\left(\frac{x}{m}\right)\\ &\sim m^{1-\gamma} \log \bar{F}(x) \sim \alpha m^{1-\gamma} \log \bar{G}(x);\end{aligned}$$

note that we compute the distribution of $mB^{(1)}$ since $B^{(1)}$ receives $1/m$ fraction of the service. Then, Theorem 6 in [20] applies with $\alpha/m^{\gamma-1} \leq \alpha$, i.e.,

$$\lim_{t\to\infty} \frac{\log \mathbb{P}[S_1^{PS} > t]}{\log t} = -\frac{\alpha}{m^{\gamma-1}}.$$

Next, we derive the *upper* bound. To this end, we consider a system where the server is kept idle after the completion of each job until the next failure occurs. At this time, all the remaining jobs are served under PS until the next shortest one departs. If there is more than one job of the same size, only one of these departs. Under this policy, it clearly holds that

$$\Theta_m^{PS} \leq \sum_{i=1}^{m} \bar{S}_i^{PS},$$

where $\bar{S}_i^{PS}$ corresponds to the service time of the ith smallest job and includes the time until the next failure.

Using the union bound, we obtain

$$\mathbb{P}[\Theta_m^{PS} > t] \le \mathbb{P}\left[\sum_{i=1}^{m} \bar{S}_i^{PS} > t\right] \le (1+\epsilon)\sum_{i=1}^{m} \mathbb{P}\left(\bar{S}_i^{PS} > \frac{t}{m}\right). \tag{5.7}$$

It is easy to see that the service time of the ith smallest job $B^{(i)}$ depends on the number of jobs that share the server, i.e., $m-i+1$, since $m-i$ jobs have remained in the queue. Now, the distribution of the ith shortest job is derived as

$$\begin{aligned}\mathbb{P}(B^{(i)} > x) &= \sum_{k=0}^{i-1} \binom{m}{k} \mathbb{P}(B_1 \le x)^k \mathbb{P}(B_1 > x)^{m-k} \\ &\sim \binom{m}{i-1} \mathbb{P}(B_1 > x)^{m-i+1} \sim \bar{F}(x)^{m-i+1}. \end{aligned} \tag{5.8}$$

Next, starting from (5.8), it easily follows that

$$\begin{aligned}\log \mathbb{P}\left((m-i+1)B^{(i)} > x\right) &\sim \log \bar{F}\left(\frac{x}{m-i+1}\right)^{m-i+1} \\ &\sim (m-i+1)^{1-\gamma} \log \bar{F}(x) \\ &\sim \alpha (m-i+1)^{1-\gamma} \log \bar{G}(x),\end{aligned}$$

where we use (5.4) and our main assumption, and define $\alpha_i \triangleq \alpha/(m-i+1)^{\gamma-1}$; here, we compute the distribution of $(m-i+1)B^{(i)}$ since the $B^{(i)}$ job receives $1/(m-i+1)$ fraction of the service.

Now, recalling Theorem 6, we have

$$\frac{\log \mathbb{P}\left[\bar{S}_i^{PS} > t\right]}{\log t} \to \alpha_i \text{ as } t \to \infty,$$

and thus (5.7) yields

$$\frac{\log \mathbb{P}\left[\Theta_m^{PS} > t\right]}{\log t} \le -(1-\epsilon) \min_{i=1,\ldots,m} \alpha_i,$$

for all $t \ge t_0$.

(i) *Subexponential or exponential jobs* ($\gamma \le 1$).
Observe that $\min_{i=1,\ldots,m} \alpha_i = \alpha$, and thus

$$\frac{\log \mathbb{P}[\Theta_m^{PS} > t]}{\log t} \le -(1-\epsilon)\alpha. \tag{5.9}$$

(ii) *Superexponential jobs* ($\gamma > 1$).
In this case, $\min_{i=1,\ldots,m} \alpha_i = \alpha/m^{\gamma-1}$, and thus

$$\frac{\log \mathbb{P}[\Theta_m^{PS} > t]}{\log t} \leq -(1-\epsilon)\frac{\alpha}{m^{\gamma-1}}. \tag{5.10}$$

Letting $\epsilon \to 0$ in (5.9) and (5.10), we obtain the upper bound.

□

6 Simulation

In this section, we present our simulation experiments in order to demonstrate our theoretical findings. All the experiments result from $N = 10^8$ (or more) samples of each simulated scenario; this guarantees the existence of at least 100 occurrences in the lightest end of the tail that is presented in the figures. First, we illustrate the instability results from Sects. 3 and 4.

Example 1 M/G/1 PS is unstable. In this example, we show that the PS queue becomes unstable by simulating the M/G/1 PS queue for different arrival rates $\lambda > 0$, which all satisfy the stability condition for the non-preemptive M/G/1 queue, when jobs are served one at a time. In this regard, we assume constant job size $\beta = 1$ and Poisson failures of rate $\mu = 1/20$. Therefore, by evaluating (3.5), we obtain

$$\lambda \mathbb{E}[S] = \lambda \mu^{-1}\left(e^{\mu} - 1\right) = 20\left(e^{0.05} - 1\right)\lambda = 1.025\lambda < 1,$$

or equivalently the stability region for the non-preemptive queue is given by $\Lambda = \{\lambda \leq 0 : \lambda < 0.9752\}$. Hence, in this example, we use λ from the preceding stability region, $\lambda \in \Lambda$.

In Fig. 3, we plot the number of jobs that have received service up to time t. We observe that the cumulative number of served jobs always converges to a fixed number and does not increase any further. This happens after some critical time when the queue starts to grow continuously and is unable to drain. For larger values of λ, the system saturates faster, meaning that the cumulative throughput at the saturated state is lower.

Furthermore, we observe from the simulation that the system behaves as if it were stable until some critical time or queue size after which it is unable to drain. From Fig. 3, we can see that the case $\lambda = 10^{-1}$ saturates at time $t = 10^6$ and the total number of served jobs reaches 10^5. Hence, the departure rate until saturation time is $10^5/10^6 = 10^{-1}$, which is exactly equal to the arrival rate $\lambda = 10^{-1}$, corresponding to the departure rate of a stable queue. This further emphasizes the importance of studying the stability of these systems since, at first glance, they may appear stable.

Figure 4 demonstrates the queue size evolution over time. Similarly to Fig. 3, we observe that for any arrival rate λ, there is a critical time after which the queue continues to grow and never empties. This time varies depending on the simulation experiment; yet, on average, we observe that the queue remains stable for longer time when λ is

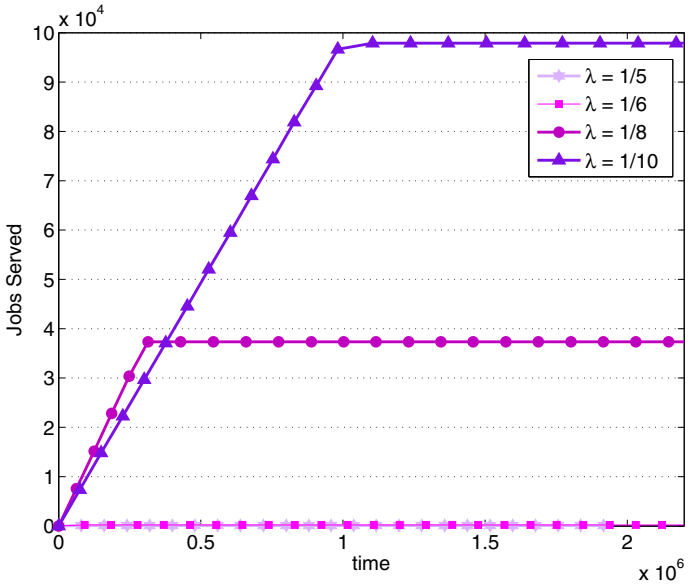
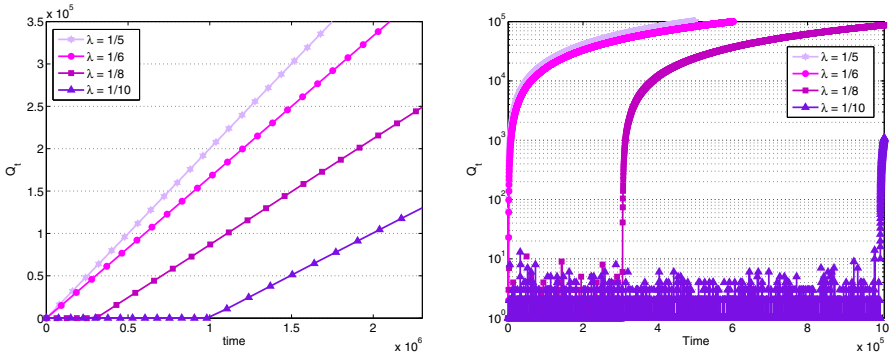


Fig. 3 Example 1. Jobs completed over time



(a) Queue size over time **(b)** Queue size for small t

Fig. 4 Example 1. **a** Queue size evolution. **b** Zooms in the time range $[0, 10^6]$ of Fig. 4; Q_t (y-axis) is shown on the logarithmic scale

smaller. Now, we zoom in on the queue evolution on the logarithmic scale in Fig. 4b. Again, we observe that the queue looks stable until some critical time/queue size.

Lastly, in Fig. 5, we plot the queue evolution for different job sizes, namely $\beta = 1, 1.2, 1.5$, and 2. We observe that larger job fragments cause instability much faster than the smaller units. For example, $\beta = 2$ leads to instability almost immediately, while $\beta = 1.5$ renders the queue unstable after 10^4 time units. Similarly, reducing the fragment size by 60 % delays the process by an additional 3×10^4 units. Lastly, cutting the jobs in half causes instability after approximately 13×10^4 time units. This implies that one should apply fragmentation with caution in order to select the appropriate fragment size that will maintain good system performance for the longest time.

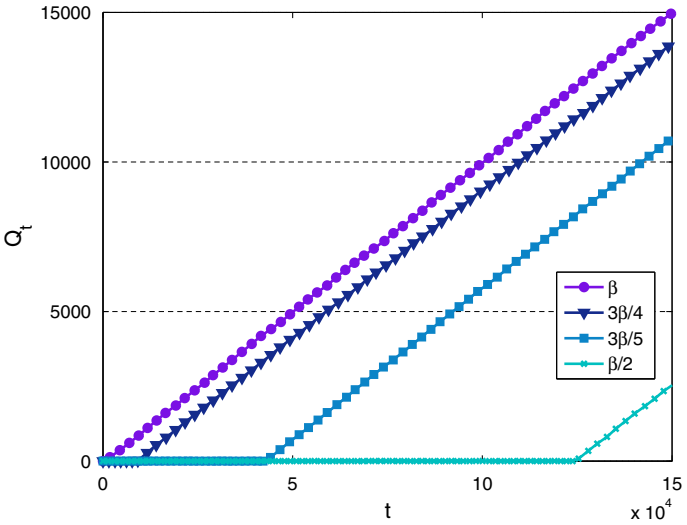


Fig. 5 Example 1. Queue size over time parameterized by fragment length; $\beta = 2, \lambda = 0.1$

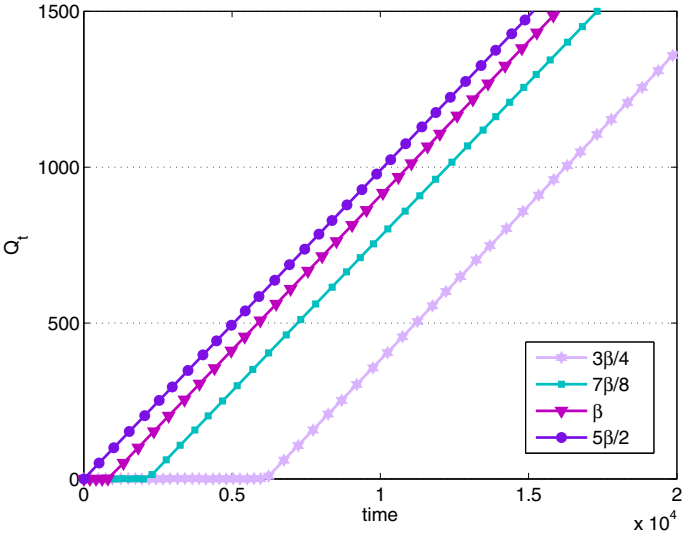


Fig. 6 Example 2. Queue size over time parameterized by job size; $\beta = 4$

Example 2 General arrivals. In this example, we consider non-Poisson arrivals. We assume that the failure distribution is exponential with mean $\mathbb{E}A = 10$ and that job interarrival times follow the Pareto distribution with $\alpha = 2$ and mean $\mathbb{E}\tau = 10.1$. Similarly to the previous example, Fig. 6 shows the queue evolution with time for different job sizes β.

Next, we validate the results on the transient analysis from Sect. 5.

Example 3 Non-preemptive policy: always the same index α. In this example, we consider a queue of $m = 10$ jobs, which are served FCFS, i.e., one at a time. The

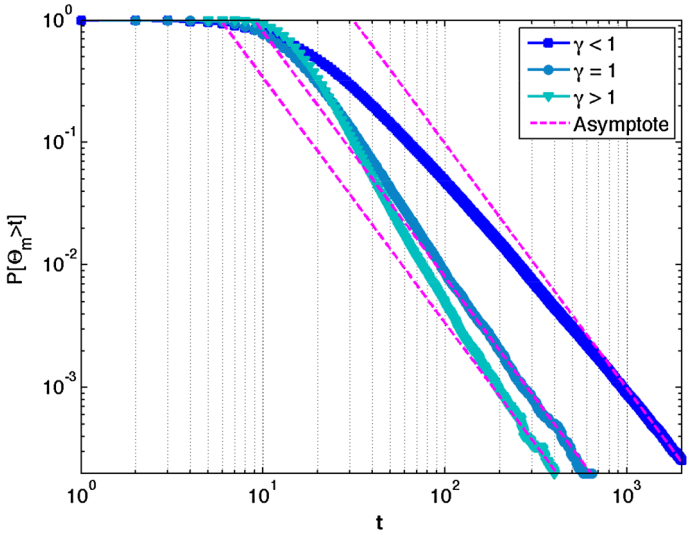


Fig. 7 Example 3. Non-preemptive policy: Logarithmic asymptotics when $\alpha = 2$ for exponential, super-exponential ($\gamma > 1$), and subexponential ($\gamma < 1$) distributions

logarithmic asymptotics from Theorem 7 imply that the tail is always a power law of index $\alpha = 2$.

In Fig. 7, we plot the distribution of the total completion time in a queue with 10 jobs that are processed one at a time. On the same graph, we plot the logarithmic asymptotics (dotted lines) that correspond to a power law of index $\alpha = 2$. We consider the following three scenarios:

1. Weibull distributions with $\gamma = 2$. The failures A are distributed according to $\bar{G}(x) = e^{-(x/\mu)^2}$ with mean $\mathbb{E}[A] = \mu\Gamma(1.5) = 1.5$, and jobs B also follow Weibull distributions with $\bar{F}(x) = e^{-(x/\lambda)^2}$, $\lambda = \mu/\sqrt{2}$. In this case, it is easy to check that the main assumption of Theorem 7 is satisfied, i.e.,

$$\log \bar{F}(x) = -(x/\lambda)^2 = \alpha \log \bar{G}(x), \quad \alpha = (\mu/\lambda)^2.$$

2. Exponential distributions. Failures are exponential with $\mathbb{E}[A] = 2$, $\bar{G}(x) = e^{-x/2}$, and the jobs B are also exponential of unit mean, i.e., $\bar{F}(x) = e^{-x}$. Then, trivially,

$$\log \bar{F}(x) = 2 \log \bar{G}(x).$$

3. Weibull distributions with $\gamma = 0.5$. Failures are Weibull with $\bar{G}(x) = e^{-\sqrt{x}/2}$, i.e., $\mathbb{E}[A] = 8$. Also, we assume Weibull jobs B with $\bar{F}(x) = e^{-\sqrt{x}}$. Thus,

$$\log \bar{F}(x) = -\sqrt{x} = 2 \log \bar{G}(x).$$

In all three cases, we obtain $\alpha = 2$. Yet, we observe that the tail asymptotics are the same regardless of the distribution of the job sizes. For the subexponential

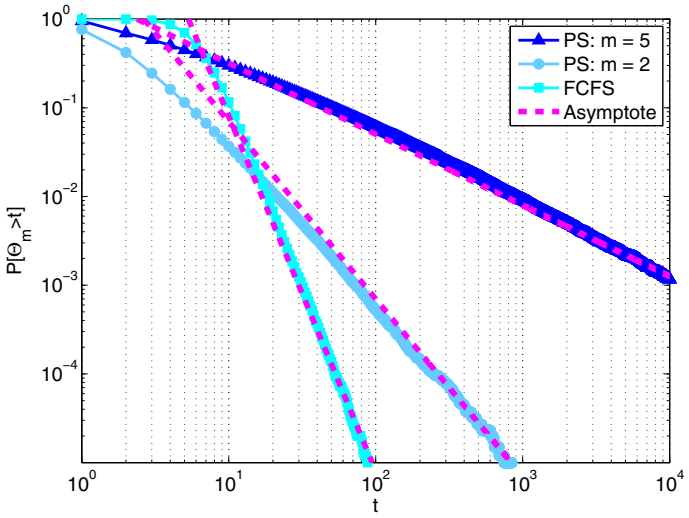


Fig. 8 Example 4. Logarithmic asymptotics for different number of superexponential jobs when $\alpha = 4$ under PS and FCFS discipline

jobs (case 3: Weibull with $\gamma < 1$), the power law tail appears later compared to the case of superexponential jobs. This is because the constant factor of the exact asymptotics are different for each case, and it depends on the mean size of A, $\mathbb{E}[A]$.

Example 4 PS: the effect of the numbers of jobs. In this example, we consider a PS queue with $m = 5$ and $m = 2$ superexponential jobs, and compare it against a FCFS queue with $m = 5$ jobs. We assume superexponential job sizes B's and A's, namely Weibull with $\gamma = 2$; see case 1 of Example 3. Here α is taken equal to 4. The logarithmic asymptotics are given in Theorems 7 and 8.

In Fig. 8, we demonstrate the total completion time Θ_m^{PS}, for different numbers of jobs, when $\gamma = 2$. Theorem 8(ii) states that $\alpha(m) = \alpha/m^{\gamma-1}$, and thus for $\gamma = 2$ we have $\alpha(m) = \alpha/m$, i.e., we expect power law asymptotes with index α/m for the different values of m. On the same figure, we also plot the FCFS completion time Θ_m, which is always a power law of index $\alpha = 4$, as we previously observed in Example 3. It can be seen that PS generates heavier power laws, for superexponential jobs. In particular, PS with $m = 2$ results in power law asymptotics with $\alpha(2) = 2$, while PS with $m = 5$ jobs leads to infinite expected delay since $\alpha(5) = 4/5 < 1$.

Example 5 PS: the effect of the distribution type. In this example, for completeness, we evaluate the impact of the job distribution on the total completion time under both heavy- and light-tailed job sizes. To this end, we consider the PS queue from Example 4, with $m = 5$ jobs, and compare it against FCFS. In Fig. 9, we re-plot the logarithmic asymptotics of the total completion time $\mathbb{P}(\Theta_m^{PS} > t)$ for different distribution types of the failures/jobs and index $\alpha = 4$, as before. In particular, we consider Weibull distributions as in Example 3 with $\gamma = 1/2 < 1$ and $\gamma = 2 > 1$ for the subexponential and superexponential cases, respectively.

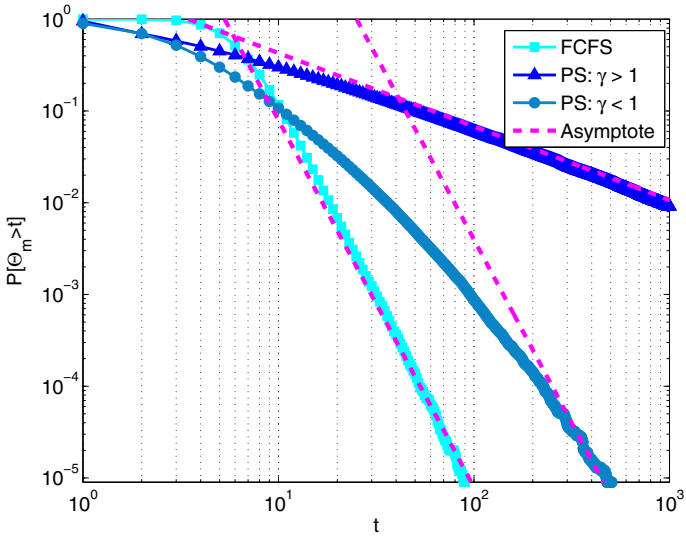


Fig. 9 Example 5. Logarithmic asymptotics under FCFS, PS with subexponential and superexponential jobs

On the same graph, we plot the distribution of the completion time Θ_m in FCFS, which is always a power law of the same index, as illustrated in Example 3. By fixing the number of jobs to be $m = 5$, Fig. 9 shows that when the jobs are superexponential, PS yields the heaviest asymptotics among all three scenarios; for subexponential jobs, PS generates asymptotics with the same power law index as in FCFS, albeit with a different constant factor.

Example 6 Limited queue: throughput versus overhead tradeoff. In practice, job and buffer sizes are bounded and therefore the queue may never become unstable. However, our results indicate that the queue may lock itself in a 'nearly unstable' state, where it is at its maximum size and the throughput is very low. Here, we would like to emphasize that, unlike in the case of unlimited queue size, job fragmentation can be useful for increasing the throughput and the efficiency of the system. In this case, one has to be careful about the overhead cost of fragmentation. Basically, each fragment requires additional information, called the 'header' in the context of communications, which contains details on how it fits into the bigger job, for example, destination/routing information in communication networks. Hence, if the fragments are too small, there will be a lot of overhead and waste of resources. In view of this fact, one would like to optimize the fragment sizes by striking a balance between throughput and utilization.

In this example, we demonstrate the tradeoff between throughput and generated overhead, assuming limited queue size q^*. If the newly arriving job does not fit in the queue, i.e., the number of jobs currently in the queue is equal to q^*, it is discarded. We define throughput as the percentage of the jobs that complete service among all jobs that arrive at the M/G/1 PS queue. It basically corresponds to the total work that is carried out in the system. On the other hand, we define utilization as the useful work that is served over the aggregated load in the system. Specifically, we consider

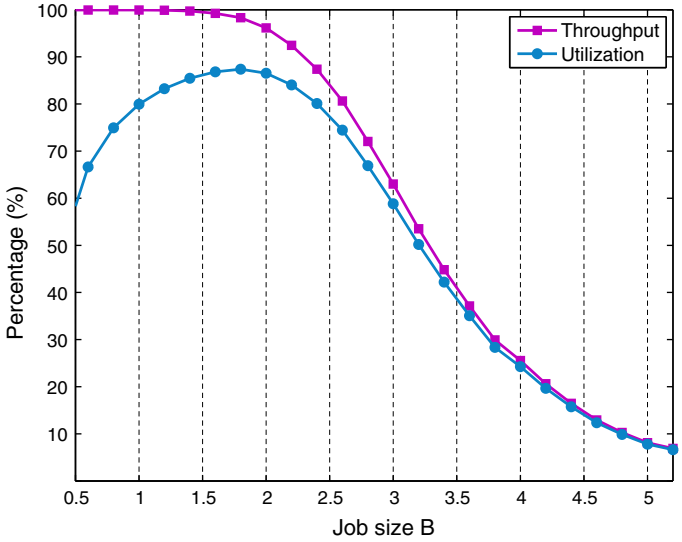


Fig. 10 Example 6. Throughput versus utilization tradeoff

jobs that require a minimum size b, where b represents the overhead, for example, the packet header, thread id, etc. The remaining job size, $\beta - b$, represents the useful information.

We consider different job sizes β from 0.4 up to 5 bytes, with overhead $b = 0.2$. We simulate the M/G/1 PS queue with maximum queue size $q^* = 10$ jobs for a fixed time $T = 10^8$ time units. The arrivals are Poisson with rate $1/10$ and the failures are exponential of the same rate. Clearly, in the case of fixed job sizes β, throughput γ is lower bounded by the throughput of the system when it performs at the limit, i.e., when the queue is full. This state corresponds to the worst overall performance and can be easily computed. On average, for a fixed period of time T, q^* jobs will complete service every $\mathbb{E}[S_{q^*}]$ time units, while the total jobs that arrive in the system is λT. In this case, the lower bound for the throughput is given by

$$\underline{\gamma} = q^* \frac{T}{\mathbb{E}[S_{q^*}]} \frac{1}{\lambda T} = \frac{q^*}{\lambda \mathbb{E}[S_{q^*}]},$$

and in the particular case of exponential failures, using (3.5) we derive

$$\underline{\gamma} = \frac{q^*}{\lambda \mu^{-1}(e^{\mu q^* \beta} - 1)}.$$

Using this observation, throughput will be suboptimal when $\gamma < 1$. Thus, for job sizes larger than $\beta_* = \log(\mu q^* \lambda^{-1} + 1)/(\mu q^*)$, the throughput starts decreasing.

In Fig. 10, we observe that for small job sizes, the throughput is 100 % and it deteriorates as the job size β increases. In particular, when the job size exceeds 1.5, the throughput drops exponentially. Utilization exhibits a different behavior; it is low when the job size is small, i.e., the useful job size is comparable to the overhead b, and

reaches its peak at $\beta \approx 1.7$. After this, it starts decreasing, following a similar trend as the throughput. In this case, $\beta - b \approx 1.5$ appears to be the optimal size for the job fragments. This phenomenon of combining limited queue size with job fragmentation may require further investigation.

7 Concluding remarks

Retransmissions/restarts represent a primary failure recovery mechanism in large-scale engineering systems, as it was argued in the introduction. In communication networks, retransmissions lie at the core of the network architecture, as they appear in all layers of the protocol stack. Similarly, PS/DPS-based scheduling mechanisms, due to their inherent fairness, are commonly used in computing and communication systems. Such mechanisms allow for efficient and fair resource allocation, and thus they are preferred in engineering system design.

However, our results show that, under mild conditions, PS/DPS scheduling in systems with retransmissions is always unstable. Furthermore, this instability cannot be resolved by job fragmentation techniques or checkpointing. On the contrary, serving one job at a time, for example, FCFS, can be stable and its performance can be further enhanced with fragmentation. Interestingly, systems where jobs are served one at a time can highly benefit from fragmentation and, in fact, their performance can approach closely the corresponding system without failures.

Overall, using PS in combination with retransmissions in the presence of failures deteriorates the system performance and induces instability. In addition, our findings suggest that further examination of existing techniques is necessary in the failure-prone environment with retransmission/restart failure recovery and sharing, for example, see Example 6.

Acknowledgments The authors are grateful to the anonymous reviewers for careful reading and helpful suggestions.

References

1. Jelenković, P.R., Skiani, E.D.: Is sharing with retransmissions causing instabilities? In: Proceedings of the The 2014 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '14), pp. 167–179. SIGMETRICS Perform. Eval. Rev. **42**(1), 167–179 (2014)
2. Bertsekas, D.P., Gallager, R.: Data Networks, 2nd edn. Prentice Hall, Englewood Cliffs (1992)
3. Yashkov, S., Yashkova, A.: Processor sharing: a survey of the mathematical theory. Autom. Remote Control **68**(9), 1662–1731 (2007)
4. Parekh, A.K., Gallager, R.G.: A generalized processor sharing approach to flow control in integrated services networks: the single-node case. IEEE/ACM Trans. Netw. **1**(3), 344–357 (1993)
5. Altman, E., Avrachenkov, K., Ayesta, U.: A survey on discriminatory processor sharing. Queueing Syst. Theory Appl. **53**(1–2), 53–63 (2006)
6. Fayolle, G., Mitrani, I., Iasnogorodski, R.: Sharing a processor among many job classes. J. ACM **27**(3), 519–532 (1980)
7. Kleinrock, L.: Time-shared systems: A theoretical treatment. J. ACM **14**(2), 242–261 (1967)
8. Coffman Jr, E.G., Muntz, R.R., Trotter, H.: Waiting time distributions for processor-sharing systems. J. ACM **17**(1), 123–130 (1970)

9. Yashkov, S.: Mathematical problems in the theory of shared-processor systems. J. Sov. Math. **58**(2), 101–147 (1992)
10. Anantharam, V.: Scheduling strategies and long-range dependence. Queueing Syst. **33**(1–3), 73–89 (1999)
11. Jelenković, P., Momčilović, P.: Large deviation analysis of subexponential waiting times in a processor sharing queue. Math. Oper. Res. **28**(3), 587–608 (2003)
12. Wierman, A., Zwart, B.: Is tail-optimal scheduling possible? Oper. Res. **60**(5), 1249–1257 (2012)
13. Fiorini, P.M., Sheahan, R., Lipsky, L.: On unreliable computing systems when heavy-tails appear as a result of the recovery procedure. SIGMETRICS Perform. Eval. Rev. **33**(2), 15–17 (2005)
14. Sheahan, R., Lipsky, L., Fiorini, P.M., Asmussen, S.: On the completion time distribution for tasks that must restart from the beginning if a failure occurs. SIGMETRICS Perform. Eval. Rev. **34**(3), 24–26 (2006)
15. Asmussen, S., Fiorini, P.M., Lipsky, L., Rolski, T., Sheahan, R.: Asymptotic behavior of total times for jobs that must start over if a failure occurs. Math. Oper. Res. **33**(4), 932–944 (2008)
16. Jelenković, P.R., Tan, J.: Can retransmissions of superexponential documents cause subexponential delays? In: Proceedings of IEEE INFOCOM'07, pp. 892–900. (2007)
17. Jelenković, P.R., Skiani, E.D.: Uniform approximation of the distribution for the number of retransmissions of bounded documents. In: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '12, pp. 101–112 (2012)
18. Jelenković, P.R., Skiani, E.D.: Distribution of the number of retransmissions of bounded documents. Adv. Appl. Prob. **47**(2), (2015). (to appear) arXiv:1210.8421
19. Jelenković, P.R., Skiani, E.D.: Retransmissions over correlated channels. SIGMETRICS Perform. Eval. Rev. **41**(2), 15–25 (2013)
20. Jelenković, P.R., Tan, J.: Characterizing heavy-tailed distributions induced by retransmissions. Adv. Appl. Probab. **45**(1): 106–138 (2013). (extended version arXiv: 0709.1138v2)