Response Time Distribution of a Class of Limited Processor Sharing Queues*

Miklos Telek
Department of Telecommunication,
Technical University of Budapest
MTA-BME Information Systems Research Group
Hungary
telek@hit.bme.hu

Benny Van Houdt
Department of Math and Computer Science,
University of Antwerp, imec
Belgium
benny.vanhoudt@uantwerpen.be

ABSTRACT

Processor sharing queues are often used to study the performance of time-sharing systems. In such systems the total service rate $\mu(m)$ depends on the number of jobs m present in the system and there is a limit implemented, called the multi-programming level (MPL), on the number of jobs k that can be served simultaneously. Prior work showed that under highly variable jobs sizes, setting the MPL k beyond the value $k^* = \arg\max_m \mu(m)$ may reduce the mean response time.

In order to study the impact of the MPL k on the response time *distribution*, we analyse the MAP/PH/LPS-k(m) queue. In such a queue jobs arrive according to a Markovian arrival process (MAP), have phase-type (PH) distributed sizes, at most k jobs are processed in parallel and the total service rate depends on the number of jobs being served. Jobs that arrive when there are k or more jobs present are queued.

We derive an expression for the Laplace transform of the response time distribution and numerically invert it to study the impact of the MPL k. Numerical results illustrate to what extent increasing k beyond k^* increases the quantiles and tail probabilities of the response time distribution. They further demonstrate that for bursty arrivals and larger MPL k values having more variable job sizes may reduce the mean response time.

CCS CONCEPTS

• **Mathematics of computing** → *Queueing theory*; • **Software and its engineering** → *Scheduling*;

KEYWORDS

Processor sharing, response time distribution, multi-programming level

ACM Reference format:
Miklos Telek and Benny Van Houdt. 2017. Response Time Distribution of a Class of Limited Processor Sharing Queues. In *Proceedings of IFIP WG. 7.3 Performance conference, New York City, USA, November 2017 (Performance'17),* 12 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

*This work was supported by the FWO grant G024514N and the OTKA 123914 project.

Performance'17, November 2017, New York City, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Processing jobs with highly variable job sizes in a first-come-first-served (FCFS) order is clearly suboptimal as short jobs get stuck behind long jobs. Therefore many systems employ some form of time sharing such that short jobs experience much better delay characteristics which often results in an overall gain in the mean system response time. In such systems the efficiency tends to improve as the number of jobs processed in parallel increases up to some point after which the efficiency declines as too many jobs are competing for the available resources [13]. In a queueing theoretic setting such a system corresponds to a processor sharing (PS) queue where the overall processing rate of the server depends on the number of jobs present. If we denote $\mu(m)$ as the processing rate in case there are m jobs served in parallel, then $\mu(m)$ is called the service curve and this curve tends to be unimodal and attains a maximum in some value k^*, i.e., $\mu(k^*) \geq \mu(m)$ for any m. As it is desirable that the system operates at high efficiency a common approach is to limit the number of jobs that is processed in parallel to some k, where k is called the multi-programming level (MPL) and setting $k = k^*$ is a very natural choice. Whenever the number of jobs n in the system exceeds k, $n - k$ of the jobs queue to receive service. Thus, the system operates as a *limited* processor sharing queue where the service rate depends on the number of jobs in service.

As indicated in [5] when the job sizes are highly variable the mean response times can be further reduced by picking an MPL k that exceeds k^* (especially if the load is not too high). The intuition is that although setting $k > k^*$ reduces the service rate when there are many jobs present, short jobs tend to have an easier time to pass long jobs which outweighs the reduced service rate (if the load is not too high). However, service disciplines that minimize the mean response time, like shortest job next, also tend to cause some form of starvation for the long jobs. Thus it would be interesting to see how the MPL k affects not only the mean response time, but the response time *distribution*. In fact it can be anticipated that increasing k beyond k^* may increase the tail probabilities as long jobs have a more difficult time to complete.

The main objective of this paper is to develop an efficient numerical method to compute the response time *distribution* in a limited processor sharing queue to investigate how it is affected by the MPL k. To this end we derive an expression for the Laplace transform

(LT) of the response time distribution of the MAP/PH/LPS-k(m) queue, where jobs arrive according to a Markovian arrival process (MAP) of order n_a, jobs have an order n_s phase-type (PH) distributed size, at most k jobs are processed in parallel and the service rate depends on the number of jobs being served. Jobs that arrive when there are k or more jobs present are queued (in an infinite buffer) and queued jobs are taken in FCFS order from the queue whenever a job completes service. To obtain the response time distribution we numerically invert the LT, which requires us to numerically evaluate the LT at various values of s. For this purpose we propose two approaches: a Kronecker and a spectral expansion approach. At first glance both approaches may appear problematic as they rely on matrix computations where the size of the matrices involved is $O(n_a n_s^k)$. However, as we are mostly interested in highly variable job sizes we focus on hyper-exponential distributions with $n_s = 2$ phases only (as in [5]). Further, it is easy (see [5]) to reduce these $O(n_a 2^k)$ size matrices to matrices of size $O(n_a k)$, which allows fast numerical inversion as indicated by the various numerical examples (for $n_a = 1, n_s = 2$ and $k = 15$ computing the probability $P[R \leq t]$ that the sojourn time is less than or equal to t for a given t requires less than 10 seconds on a regular laptop).

The main insights provided by the numerical examples is that while increasing the MPL k beyond k^* reduces the mean response times in case of highly variable job sizes, it does significantly increase the tail probabilities. More surprisingly, when the arrivals occur in bursts and the MPL k is large (e.g., $k = 15$) more variable job sizes can result in significantly lower mean response times (while the tail probabilities still increase with increasing job size variability).

The paper is structured as follows. The queueing model analyzed in this paper is described in Section 2, while Section 3 reviews some of the related work in this area. Section 4 focuses on how to compute the queue length distribution. The analysis of the response time distribution is presented in Section 5 and is the main technical contribution of the paper. Section 6 discusses various numerical examples and conclusions are drawn in Section 7.

2 MODEL AND NOTATIONS

To model general service time distribution and job arrival patterns we assume that customers arrive according to a Markovian arrival process (MAP) and their job lengths are phase-type (PH) distributed. PH distributions and MAPs are distributions and point processes with a modulating finite state background Markov chain [7]. At the expense of increasing the size of the background Markov chain any general distribution and point process can be approximated arbitrary closely with PH distributions and MAPs, respectively. MAPs can describe point processes with possibly dependent inter-arrival times and include the special case of renewal processes with PH distributed inter-arrival times. Various fitting tools are available online for PH distributions, as well as some tools to approximate MAPs (e.g., [6]).

We consider an infinite buffer service node, where jobs arrive according to a MAP$(\mathbf{D_0}, \mathbf{D_1})$ with mean arrival rate $\lambda = \delta \mathbf{D_1} \mathbb{1}$, where $\mathbb{1}$ is the column vector of ones of appropriate size and δ is the stationary distribution of the MAP satisfying $\delta(\mathbf{D_0}+\mathbf{D_1}) = 0$ and $\delta\mathbb{1} = 1$. The service time distribution is PH$(\alpha, \mathbf{A})$. The completion rate vector of PH$(\alpha, \mathbf{A})$ is denoted by $\mathbf{a} = -\mathbf{A}\mathbb{1}$. The size of the arrival MAP is n_a and the size of the service PH is n_s.

The server serves at most k jobs in parallel such that the service speed depends on the number of parallel served jobs. If m ($m \leq k$) jobs are served in parallel the service speed of one of those jobs is $\mu_m = \mu(m)/m$ and during such a period the service time is PH$(\alpha, \mu_m \mathbf{A})$. We only consider MPL values k for which the stability condition

$$\lambda\alpha(-\mathbf{A})^{-1}\mathbb{1} < k\mu_k,$$

holds.

We are interested in the response time distribution of this system. In order to compute it we first compute the stationary distribution of the number of customers, the service phases and the phase of the arrival process, based on which we evaluate the distribution observed by a tagged customer arriving to the system. This part of the analysis, presented in Section 4, is a straightforward application of matrix analytic methods [7, 10]. The challenge is to determine the response time distribution given the initial distribution of a tagged customer in an efficient manner: this is by far the main technical contribution of the paper and is presented in Section 5. The response time distribution is obtained in the Laplace transform domain from which a numerical inverse transformation provides the time domain result.

3 RELATED WORK

The closest related work is [5] which considers exactly the same queueing system, but focuses on the mean response time only. As the MPL that minimizes the mean response time depends to a large extent on the arrival rate λ the authors also propose two dynamic schemes to adjust the MPL level to minimize the mean response time. In [12] a numerical scheme is proposed to study the response time distribution of a limited processor sharing queue, but the system is a closed queueing system and job durations are exponential. While the analysis can be adapted to an open queueing network with Poisson input, relaxing the exponential job durations appears problematic.

Limited processor sharing (LPS) queues (where the overall service rate does not depend on the number of jobs being processed) have been studied by a variety of authors. In [9] the impact of the MPL k on the sojourn time tail asymptotics is studied and a robust setting for k is proposed that achieves good tail asymptotics for both heavy and light tailed job size distributions. A method to assess the mean response time in an LPS queue (with Poisson arrivals and PH service) via matrix analytic methods was proposed in [14] and is very similar in nature to Section 4. A closed form approximation for the mean sojourn time in an LPS queue with general service times was proposed in [2]. Fluid, diffusion and heavy traffic approximations for LPS queues have also been developed in [15–17]. Further, a monotonicity result for the G/GI/LPS-k queue was presented in [11], which shows that the queue length distribution monotonically increases (decreases) in the stochastic ordering sense as a function of the MPL k for service time distributions with an increasing (decreasing) hazard rate.

4 STATIONARY ANALYSIS OF THE NUMBER OF CUSTOMERS

Let $N(t)$ be the number of customers in the system at time t, $Z(t)$ be the phase of the MAP characterizing the arrivals at time t and let $\{Y_1(t), Y_2(t), \ldots, Y_{\min(N(t),k)}(t)\}$ represent the service phases of the customers under service at time t. Clearly, $X(t) = (N(t), \{Z(t), Y_1(t), Y_2(t), \ldots, Y_{\min(N(t),k)}(t)\})$ is a continuous time Markov process. We state that $N(t)$ is the "level" process and $\{Z(t), Y_1(t), Y_2(t), \ldots, Y_{\min(N(t),k)}(t)\}$ is the "phase" process. Denote the stationary probabilities of level m by π_m, i.e., π_m is a row vector containing the stationary probabilities for $(N(t), \{Z(t), Y_1(t), Y_2(t), \ldots, Y_{\min(m,k)}(t)\})$ with $N(t) = m$. Define the Kronecker sum for vectors $\oplus_{i=1}^{m}\mathbf{a} \triangleq \sum_{i=1}^{m} \otimes_{j=1}^{i-1}\mathbf{I} \otimes \mathbf{a} \otimes \otimes_{\ell=i+1}^{m}\mathbf{I}$, then we have the following result:

Theorem 4.1. *Let* $\mathbf{L_m} = \mathbf{D_0} \oplus \oplus_{i=1}^{m}\mu_m\mathbf{A}$ *and* $\mathbf{F_m} = \mathbf{D_1} \otimes \alpha \otimes \otimes_{i=1}^{m}\mathbf{I}$, *for* $m = 0, \ldots, k-1$. *Let* $\mathbf{B_m} = \mathbf{I} \otimes \oplus_{i=1}^{m}\mu_m\mathbf{a}$, *for* $m = 1, \ldots, k$, $\mathbf{L} = \mathbf{L_k}$, $\mathbf{F} = \mathbf{D_1} \otimes \otimes_{i=1}^{k}\mathbf{I}$, $\mathbf{B} = \mathbf{I} \otimes \oplus_{i=1}^{k}\mu_k\mathbf{a}\alpha$ *and* $\mathbf{B}_{k+1} = \mathbf{B}$. *Denote* $\mathbf{R}$ *as the minimal non-negative solution of the quadratic matrix equation*

$$\mathbf{0} = \mathbf{F} + \mathbf{RL} + \mathbf{R}^2\mathbf{B}, \tag{1}$$

$\mathbf{R_k} = \mathbf{R}$ *and define*

$$\mathbf{R_m} = -\mathbf{F_m}(\mathbf{L_{m+1}} + \mathbf{R_{m+1}B_{m+2}})^{-1}, \tag{2}$$

for $m = 0, \ldots, k-1$, *then*

$$\pi_m = \pi_0 \prod_{j=0}^{m-1} \mathbf{R_j},$$

$$\pi_{k+j} = \pi_k \mathbf{R}^j,$$

for $m = 0, \ldots, k$ *and* $j \geq 0$ *and* π_0 *is the unique vector such that* $\pi_0(\mathbf{L_0} + \mathbf{R_0B_1}) = 0$ *and* $\sum_i \pi_i \mathbb{1} = 1$.

Proof. Due to the fact that $N(t)$ can change at most by one in a single state transition $X(t)$ is a Quasi-Birth-Death (QBD) Markov process, where the first $k+1$ levels $(0, 1, \ldots, k)$ of this QBD are irregular and level dependent. The generator of the QBD is given by

$$\begin{bmatrix}
\mathbf{L_0} & \mathbf{F_0} & & & & & & \\
\mathbf{B_1} & \mathbf{L_1} & \mathbf{F_1} & & & & & \\
 & \ddots & \ddots & \ddots & & & & \\
 & & \mathbf{B_{k-1}} & \mathbf{L_{k-1}} & \mathbf{F_{k-1}} & & & \\
 & & & \mathbf{B_k} & \mathbf{L} & \mathbf{F} & & \\
 & & & & \mathbf{B} & \mathbf{L} & \mathbf{F} & \\
 & & & & & \ddots & \ddots & \ddots
\end{bmatrix},$$

The result follows from a standard application of matrix analytic methods [7, 10]. □

The matrices $\mathbf{R_m}$ are of size $n_a n_s^m \times n_a n_s^{m+1}$ and $\mathbf{R}$ is a square matrix of size $n_a n_s^k$. While this may appear problematic for computational purposes, we can use the common method to reduce the size of the matrices by simply keeping track of the number of servers that are currently in each service phase (at the expense of complicating the expressions for the blocks part of the generator matrix), see Section 6 for more details. When $n_s = 2$, as in our numerical experiments, this makes the block sizes linear in k (as opposed to exponential).

Theorem 4.2. *After an arrival of a stationary customer the system state distribution is*

$$\hat{\pi}_{m+1} = \frac{1}{\lambda}\pi_m\mathbf{F_m},$$

where $\mathbf{F_m} = \mathbf{F}$ *for* $m \geq k$.

Proof. The probability of being in phase u of level $m+1$ just after an arrival epoch is $(\pi_m\mathbf{F_m})_u$ and the result then follows by a standard Markov chain argument [3]. It suffices to check that

$$\sum_{m=0}^{\infty} \hat{\pi}_{m+1}\mathbb{1} = \sum_{m=0}^{\infty} \frac{1}{\lambda}\pi_m\mathbf{F_m}\mathbb{1} =$$

$$\sum_{m=0}^{\infty} \frac{1}{\lambda}\pi_m\mathbf{D_1}\mathbb{1} \otimes 1 \otimes \otimes_{i=1}^{min(m,k)}\mathbb{1} = \frac{1}{\lambda}\delta\mathbf{D_1}\mathbb{1} = 1.$$

□

5 RESPONSE TIME DISTRIBUTION

The Laplace transform of the response time, R, is denoted by $r(s) = \mathbb{E}(e^{-sR})$. To compute the response time we introduce the following quantities.

Let entry (u, v) of the matrix $\mathbf{W}(s, i, j)$ be the Laplace transform of the probability that i customers are served and j arrive in an interval of length t where the phase process starts in phase u and ends in phase v, while service is performed by k servers during the entire interval. More precisely,

$$\bar{\mathbf{W}}(t, i, j)_{u,v} = \mathrm{P}(N_{ser}(t) = i, N_{arr}(t) = j, N(\tau) \geq k, \forall \tau < t,$$
$$\{Z(t), Y_1(t), \ldots, Y_k(t)\} = v | \{Z(0), Y_1(0), \ldots, Y_k(0)\} = u),$$

where $N_{arr}(t)$ and $N_{ser}(t)$ denote the number of arrivals and services in $(0, t)$, respectively, and $\mathbf{W}(s, i, j) = \int_t e^{-st}\bar{\mathbf{W}}(t, i, j)dt$. As the elements of $\bar{\mathbf{W}}(t, i, j)$ are probabilities, for $Re(s) > 0$ we have

$$\int_0^\infty \left|e^{-st}\bar{\mathbf{W}}(t, i, j)_{u,v}\right| dt \leq \int_0^\infty \left|e^{-st}\right| dt = \int_0^\infty e^{-Re(s)t}dt < \infty,$$

and consequently $\mathbf{W}(s, i, j)_{u,v}$ is analytic for $Re(s) > 0$ (as the LT is analytic in its absolute region of convergence). In the subsequent analysis we present Laplace domain matrix expressions for matrices of probabilities, these expressions are obtained as the solution of linear equations in explicit (described by matrix inverse) or in implicit (solution of a Sylvester equation) from. In either cases, without any further notice, we have that the solution is finite and analytic for $Re(s) > 0$ due to the same reasoning as for $\mathbf{W}(s, i, j)_{u,v}$.

Define entry u of the column vector $\mathbf{S_i}(s)$ as the Laplace transform of the remaining service time R_{rem} of a tagged customer when there are i customers in the system, the phase process is in phase u and the tagged customer is in the first server. That is,

$$\mathbf{S_i}(s)_u = \mathbb{E}(e^{-sR_{rem}} | N(0) = i, \text{tagged in server 1},$$
$$\{Z(0), Y_1(0), \ldots, Y_{\min(i,k)}(0)\} = u).$$

Theorem 5.1. *The Laplace transform of the response time,* $r(s)$, *can be expressed as*

$$r(s) = \sum_{m=0}^{k-1} \frac{1}{\lambda}\pi_m\mathbf{F_m}\mathbf{S_{m+1}}(s) + \sum_{j=0}^{\infty}\sum_{i=0}^{\infty} \frac{1}{\lambda}\pi_k\mathbf{R}^i\mathbf{F}\mathbf{W}(s, i, j)\mathbf{B'}\mathbf{S_{k+j}}(s) \tag{3}$$

where $\mathbf{B'} = (\mathbf{I} \otimes \oplus_{i=1}^{k}\mu_k\mathbf{a})(\mathbf{I} \otimes \alpha \otimes_{i=1}^{k}\mathbf{I})$.

Proof. We rely on Theorem 4.2 and distinguish between two cases: the tagged customer either gets to the server directly, or it is buffered upon its arrival. In the first case the matrix $\mathbf{F_m}$ ensures that the newly arrived customer is in server one and its service time is described by $\mathbf{S_{m+1}}(s)$. In the second case $\frac{1}{\lambda}\pi_k \mathbf{R}^i \mathbf{F}$ describes the probability that there are $k+i$ customers in front of the tagged one and $\sum_{j=0}^{\infty} \mathbf{W}(s,i,j)$ describes all arrival patterns until i customers have departed and the tagged customers starts service. The matrix $\mathbf{B'}$ ensures that the tagged customer gets to server one, from which point $\mathbf{S_{k+j}}(s)$ describes the response time. □

Theorem 5.2. *The matrices* $\mathbf{W}(s,i,j)$ *satisfy the following relations:*

$$\mathbf{L}(s) \triangleq \mathbf{W}(s,0,0) = (s\mathbf{I} - \mathbf{L})^{-1}, \tag{4}$$

$$\mathbf{W}(s,i,0) = \mathbf{W}(s,i-1,0)\mathbf{B}\mathbf{L}(s) = (\mathbf{L}(s)\mathbf{B})^i \mathbf{L}(s) = \mathbf{L}(s)\left(\mathbf{B}\mathbf{L}(s)\right)^i, \tag{5}$$

$$\mathbf{W}(s,0,j) = \mathbf{W}(s,0,j-1)\mathbf{F}\mathbf{L}(s) = (\mathbf{L}(s)\mathbf{F})^j \mathbf{L}(s) = \mathbf{L}(s)\left(\mathbf{F}\mathbf{L}(s)\right)^j, \tag{6}$$

$$\mathbf{W}(s,i,j) = \left(\mathbf{W}(s,i,j-1)\mathbf{F} + \mathbf{W}(s,i-1,j)\mathbf{B}\right)\mathbf{L}(s). \tag{7}$$

Proof. (4) describes the case without any arrivals or service completions. In this case only the phase can change according to the matrix $\mathbf{L}$. (5) represents the case that there are $i-1$ services and no arrivals in $(0,\tau)$ with $\tau < t$, there is a service completion at time τ according to matrix $\mathbf{B}$ and there are no arrivals or service completions in (τ,t). (6) is similar to (5), and (7) considers the cases that the last event is an arrival or a service completion. □

5.1 Analysis of $\mathbf{S_i}(s)$

To compute $\mathbf{S_i}(s)$ we need matrices that distinguish between a service completion of the tagged customer or another customer. To this end we define $\hat{\mathbf{B}} = \mathbf{I}\otimes\mu_k\mathbf{a}\alpha\otimes\otimes_{\ell=2}^{k}\mathbf{I}$ and $\hat{\mathbf{B}}_\mathbf{m} = \mathbf{I}\otimes\mu_m\mathbf{a}\otimes\otimes_{\ell=2}^{m}\mathbf{I}$ for $m \le k$ for the service completion of the tagged customer (residing in server one). While $\bar{\mathbf{B}} = \mathbf{I}\otimes\mathbf{I}\otimes\oplus_{\ell=2}^{k}\mu_k\mathbf{a}\alpha$ and $\bar{\mathbf{B}}_\mathbf{m} = \mathbf{I}\otimes\mathbf{I}\otimes\oplus_{\ell=2}^{m}\mu_m\mathbf{a}$ for $m \le k$ correspond to another customer completing service. Note that $\hat{\mathbf{B}} + \bar{\mathbf{B}} = \mathbf{B}$ and $\hat{\mathbf{B}}_\mathbf{m} + \bar{\mathbf{B}}_\mathbf{m} = \mathbf{B}_\mathbf{m}$. Furthermore, define $\mathbf{L_m}(s) \triangleq (s\mathbf{I} - \mathbf{L_m})^{-1}$ for $m < k$ and $\mathbf{F'_m} = \mathbf{D_1} \otimes \otimes_{i=1}^{m}\mathbf{I} \otimes \alpha$ which is the same as $\mathbf{F_m}$ except that it puts the incoming customer in server $m+1$ (instead of server one).

Theorem 5.3. $\mathbf{S_i}(s)$ *satisfies the following recurrence relations*

$$\mathbf{S_1}(s) = \mathbf{L_1}(s)\left(\hat{\mathbf{B}}_1\mathbb{1} + \mathbf{F'_1}\mathbf{S_2}(s)\right), \tag{8}$$

$$\mathbf{S_m}(s) = \mathbf{L_m}(s)\left(\hat{\mathbf{B}}_\mathbf{m}\mathbb{1} + \bar{\mathbf{B}}_\mathbf{m}\mathbf{S_{m-1}}(s) + \mathbf{F'_m}\mathbf{S_{m+1}}(s)\right), \text{ for } 1 < m < k, \tag{9}$$

$$\mathbf{S_k}(s) = \mathbf{L}(s)\left(\hat{\mathbf{B}}_\mathbf{k}\mathbb{1} + \bar{\mathbf{B}}_\mathbf{k}\mathbf{S_{k-1}}(s) + \mathbf{F}\mathbf{S_{k+1}}(s)\right), \tag{10}$$

$$\mathbf{S_{k+j}}(s) = \mathbf{L}(s)\left(\hat{\mathbf{B}}\mathbb{1} + \bar{\mathbf{B}}\mathbf{S_{k+j-1}}(s) + \mathbf{F}\mathbf{S_{k+j+1}}(s)\right), \text{ for } j \ge 1. \tag{11}$$

Proof. We detail only (9) as the other cases can be derived similarly. The response time associated with $\mathbf{S_m}(s)$ for $1 < m < k$ can be computed assuming that there are only phase transitions according to matrix $\mathbf{L_m}$ up to time τ ($\tau < t$) and at time τ there are three possible events:

- the service completion of the tagged customer according to the vector $\hat{\mathbf{B}}_\mathbf{m}\mathbb{1}$;
- the service completion of a non-tagged customer according to the matrix $\bar{\mathbf{B}}_\mathbf{k}$ from which the remaining response time is $\mathbf{S_{k-1}}(s)$;
- an arrival captured by $\mathbf{F'_m}$ (which keeps the tagged customer in server one and places the incoming one to serve $m+1$) from which the remaining response time is $\mathbf{S_{m+1}}(s)$.

□

To avoid the infinite recurrence relation in (11) we compute $\mathbf{S_{k+1}}(s)$ explicitly, based on $\mathbf{V}(s,i,h)$. Entry (u,v) of $\mathbf{V}(s,i,h)$ is the Laplace transform of the probability that i **non-tagged** customers are served and h customers arrive in an interval of length t that starts in phase u with $k+1$ customers and ends in phase v, while there are at least $k+1$ customers in the system during the entire interval of length t. There are two differences between $\mathbf{V}(s,i,h)$ and $\mathbf{W}(s,i,h)$. The first one is that $\mathbf{W}(s,i,h)$ considers the service of all customers while $\mathbf{V}(s,i,h)$ considers the service of the non-tagged customers only and assumes that the tagged customer is not served in $(0,t)$. The second (and more intricate) difference is that $\mathbf{W}(s,i,h)$ does not put any restrictions on the order of the i service completions and h arrivals, while for $\mathbf{V}(s,i,h)$ the number of arrived customers may not be below the number of served customers at all times, ensuring that there are at least $k+1$ customers in the system during the entire interval.

Theorem 5.4. *For* $\mathbf{V}(s,i,h)$ *we have*

$$\mathbf{V}(s,0,0) = (s\mathbf{I} - \mathbf{L})^{-1} = \mathbf{L}(s), \tag{12}$$

$$\mathbf{V}(s,0,h) = \mathbf{V}(s,0,h-1)\mathbf{F}\mathbf{L}(s) = (\mathbf{L}(s)\mathbf{F})^h \mathbf{L}(s), \tag{13}$$

$$\mathbf{V}(s,i,i) = \mathbf{V}(s,i-1,i)\bar{\mathbf{B}}\mathbf{L}(s), \tag{14}$$

$$\mathbf{V}(s,i,h) = \left(\mathbf{V}(s,i,h-1)\mathbf{F} + \mathbf{V}(s,i-1,h)\bar{\mathbf{B}}\right)\mathbf{L}(s), \text{ for } 0 < i < h. \tag{15}$$

Proof. The reasoning is similar to the proof of Theorem 5.2 and we only emphasize the differences here. The matrix $\bar{\mathbf{B}}$ corresponds to the service completion of a non-tagged customer, and (14) ensures that the state when the number of served and arrived customers is identical, is reachable only from the state when the number of arrivals equals the number of service completions plus one. This ensures that there cannot be fewer than $k+1$ customers in the system. □

Theorem 5.5. *For* $Re(s) \ge 0$, $\mathbf{S_{k+1}}(s)$ *can be computed as*

$$\mathbf{S_{k+1}}(s) = \mathbf{V}(s,0)\left(\mathbf{I} - \bar{\mathbf{R}}(s)\right)^{-1}\hat{\mathbf{B}}\mathbb{1} + \mathbf{V}(s,0)\bar{\mathbf{B}}\mathbf{S_k}(s), \tag{16}$$

where $\mathbf{V}(s,0) = \left(s\mathbf{I} - \mathbf{L} - \bar{\mathbf{R}}(s)\bar{\mathbf{B}}\right)^{-1}$ *and* $\bar{\mathbf{R}}(s)$ *is the minimal non-negative solution of*

$$s\bar{\mathbf{R}}(s) = \mathbf{F} + \bar{\mathbf{R}}(s)\mathbf{L} + \bar{\mathbf{R}}^2(s)\bar{\mathbf{B}}. \tag{17}$$

Proof. According to the definition of $\mathbf{V}(s,i,h)$ we have

$$\mathbf{S_{k+1}}(s) = \sum_{m=0}^{\infty}\sum_{\ell=0}^{\infty}\mathbf{V}(s,\ell,\ell+m)\hat{\mathbf{B}}\mathbb{1} + \sum_{\ell=0}^{\infty}\mathbf{V}(s,\ell,\ell)\bar{\mathbf{B}}\mathbf{S_k}(s), \tag{18}$$

where the first term describes the case when the tagged customer is served before the queue length decreases to k and the second term describes the case when the tagged customer is not served

while the queue length is larger than k and from the point when the queue length reduces to k is its remaining service time is described by $\mathbf{S_k}(s)$.

To avoid infinite summations we introduce $\mathbf{V}(s,i) = \sum_{\ell=0}^{\infty}\mathbf{V}(s,\ell,i+\ell)$ for which the following recurrence relations hold for $i>0$ based on (15)

$$\mathbf{V}(s,i) = \mathbf{V}(s,i-1)\mathbf{F}\mathbf{L}(s) + \mathbf{V}(s,i+1)\bar{\mathbf{B}}\mathbf{L}(s).$$

Substituting $\mathbf{L}(s)$ by $(s\mathbf{I}-\mathbf{L})^{-1}$ gives

$$s\mathbf{V}(s,i) = \mathbf{V}(s,i-1)\mathbf{F} + \mathbf{V}(s,i)\mathbf{L} + \mathbf{V}(s,i+1)\bar{\mathbf{B}},$$

which indicates that $\mathbf{V}(s,i)$ follows a matrix geometric sequence

$$\mathbf{V}(s,i) = \mathbf{V}(s,i-1)\bar{\mathbf{R}}(s) = \mathbf{V}(s,0)\bar{\mathbf{R}}^i(s),$$

where $\bar{\mathbf{R}}(s)$ is the solution of (17). For $Re(s)\geq 0$, the spectral radius of $\bar{\mathbf{R}}(s)$ is less than 1 since $(\mathbf{F}+\mathbf{L}+\bar{\mathbf{B}})\,\mathbb{1} = -\hat{\mathbf{B}}\mathbb{1} \leq 0$, which implies that $\left(\mathbf{I}-\bar{\mathbf{R}}(s)\right)^{-1}$ exists. For $\mathbf{V}(s,0)$ we have due to (14)

$$\begin{aligned}\mathbf{V}(s,0) &= \mathbf{V}(s,0,0) + \sum_{\ell=1}^{\infty}\mathbf{V}(s,\ell,\ell) = \mathbf{L}(s) + \sum_{\ell=1}^{\infty}\mathbf{V}(s,\ell-1,\ell)\bar{\mathbf{B}}\mathbf{L}(s)\\ &= \mathbf{L}(s) + \mathbf{V}(s,1)\bar{\mathbf{B}}\mathbf{L}(s),\end{aligned}$$

from which $\mathbf{V}(s,0)$ can be computed as

$$\mathbf{V}(s,0) = \mathbf{L}(s) + \mathbf{V}(s,0)\bar{\mathbf{R}}(s)\bar{\mathbf{B}}\mathbf{L}(s) = \left(s\mathbf{I}-\mathbf{L}-\bar{\mathbf{R}}(s)\bar{\mathbf{B}}\right)^{-1}.$$

Finally, from (18) we have

$$\begin{aligned}\mathbf{S_{k+1}}(s) &= \sum_{m=0}^{\infty}\mathbf{V}(s,0)\bar{\mathbf{R}}^m(s)\hat{\mathbf{B}}\mathbb{1} + \mathbf{V}(s,0)\bar{\mathbf{B}}\mathbf{S_k}(s),\\ &= \mathbf{V}(s,0)\left(\mathbf{I}-\bar{\mathbf{R}}(s)\right)^{-1}\hat{\mathbf{B}}\mathbb{1} + \mathbf{V}(s,0)\bar{\mathbf{B}}\mathbf{S_k}(s), \qquad (19)\end{aligned}$$

which completes the theorem. □

Thanks to (16), (10) can be written as

$$\mathbf{S_k}(s) = \mathbf{L}(s)\left(\hat{\mathbf{B}}_\mathbf{k}\mathbb{1} + \bar{\mathbf{B}}_\mathbf{k}\mathbf{S_{k-1}}(s) + \mathbf{F}\mathbf{V}(s,0)\left(\left(\mathbf{I}-\bar{\mathbf{R}}(s)\right)^{-1}\hat{\mathbf{B}}\mathbb{1} + \bar{\mathbf{B}}\mathbf{S_k}(s)\right)\right). \qquad (20)$$

(20) together with (8) and (9) form a set of linear equations for $\mathbf{S_m}(s)$ $(m=1,\ldots,k)$ for any fixed value of s.

Theorem 5.6. *For $j>0$, $\mathbf{S_{k+j}}(s)$ can be computed as*

$$\mathbf{S_{k+j}}(s) = \sum_{\ell=0}^{j-1}\left(\mathbf{V}(s,0)\bar{\mathbf{B}}\right)^{\ell}\mathbf{V}(s,0)\left(\mathbf{I}-\bar{\mathbf{R}}(s)\right)^{-1}\hat{\mathbf{B}}\mathbb{1} + \left(\mathbf{V}(s,0)\bar{\mathbf{B}}\right)^{j}\mathbf{S_k}(s), \qquad (21)$$

Proof. The recurrence relation in (16) remains valid for higher number of customers as well, that is

$$\mathbf{S_{k+j}}(s) = \mathbf{V}(s,0)\left(\mathbf{I}-\bar{\mathbf{R}}(s)\right)^{-1}\hat{\mathbf{B}}\mathbb{1} + \mathbf{V}(s,0)\bar{\mathbf{B}}\mathbf{S_{k+j-1}}(s). \qquad (22)$$

Successive substitution of this recurrence relation for $k+1,\ldots,k+j$ yields (21). □

5.2 Computing the response time distribution

According to (3) and (21) the Laplace transform of the response time is

$$\begin{aligned}r(s) &= \underbrace{\sum_{m=0}^{k-1}\frac{1}{\lambda}\pi_m\mathbf{F_m}\mathbf{S_{m+1}}(s)}_{r_1(s)} + \sum_{j=0}^{\infty}\sum_{i=0}^{\infty}\frac{1}{\lambda}\pi_k\mathbf{R}^i\mathbf{F}\mathbf{W}(s,i,j)\mathbf{B}'\mathbf{S_{k+j}}(s)\\ &= r_1(s) + \underbrace{\sum_{j=0}^{\infty}\sum_{i=0}^{\infty}\frac{1}{\lambda}\pi_k\mathbf{R}^i\mathbf{F}\mathbf{W}(s,i,j)\mathbf{B}'\left(\mathbf{V}(s,0)\bar{\mathbf{B}}\right)^j\mathbf{S_k}(s)}_{r_2(s)}\\ &+ \underbrace{\sum_{j=1}^{\infty}\sum_{i=0}^{\infty}\frac{1}{\lambda}\pi_k\mathbf{R}^i\mathbf{F}\mathbf{W}(s,i,j)\mathbf{B}'\sum_{\ell=0}^{j-1}\left(\mathbf{V}(s,0)\bar{\mathbf{B}}\right)^{\ell}\mathbf{V}(s,0)\left(\mathbf{I}-\bar{\mathbf{R}}(s)\right)^{-1}\hat{\mathbf{B}}\mathbb{1}}_{r_3(s)}\end{aligned}$$

Using Theorem 4.1 we can compute π_m $(m=1,\ldots,k)$ and from the linear system of (8), (9) and (20) we can retrieve $\mathbf{S_m}(s)$ $(m=1,\ldots,k)$ for any fixed value of s. Based on these, we can compute the first term, $r_1(s)$. We now focus on the analysis of $r_2(s)$ and $r_3(s)$, using two different computational approaches.

5.2.1 Computing the response time by Kronecker expansion. The first approach to compute $r_2(s)$ and $r_3(s)$ exists in relying on a Kronecker expansion. While the solution is easy to implement, the size of the matrices involved is the square of the size of the blocks characterizing the generator of the Markov chain. This implies that the run time complexity when $n_s=2$ will grow as k^6. The spectral approach presented in the next section avoids the need to work with such large matrices and has a time complexity that is cubic in k when $n_s=2$ (see Section 6 for more details on the run time complexity).

Theorem 5.7.

$$r_2(s) = \frac{1}{\lambda}\left(\mathbf{S_k}(s)^T\otimes\pi_k\right)\mathbf{N}(s)vec(\mathbf{M}(s,0))$$

where $\otimes$ is Kronecker product, ^{T} denotes transpose, $vec()$ is the column stacking vector operator,

$$\mathbf{U}(s) = \left(\mathbf{I}-\left(\mathbf{L}(s)^T\mathbf{B}^T\otimes\mathbf{R}\right)\right)^{-1}\left(\mathbf{L}(s)^T\mathbf{F}^T\otimes\mathbf{I}\right)$$

and $\mathbf{M}(s,0)$ and $\mathbf{N}(s)$ are the solutions of the Sylvester equations[1]

$$\mathbf{M}(s,0) = \mathbf{F}\mathbf{L}(s) + \mathbf{R}\mathbf{M}(s,0)\mathbf{B}\mathbf{L}(s),$$

$$\mathbf{N}(s) = \left(\mathbf{B}'^T\otimes\mathbf{I}\right) + \left(\bar{\mathbf{B}}^T\mathbf{V}(s,0)^T\otimes\mathbf{I}\right)\mathbf{N}(s)\mathbf{U}(s).$$

Proof. Let $\mathbf{M}(s,j) = \sum_{i=0}^{\infty}\mathbf{R}^i\mathbf{F}\mathbf{W}(s,i,j)$. For $j=0$, we have

$$\begin{aligned}\mathbf{M}(s,0) &= \sum_{i=0}^{\infty}\mathbf{R}^i\mathbf{F}\mathbf{W}(s,i,0) = \sum_{i=0}^{\infty}\mathbf{R}^i\mathbf{F}\mathbf{L}(s)\left(\mathbf{B}\mathbf{L}(s)\right)^i\\ &= \mathbf{F}\mathbf{L}(s) + \mathbf{R}\mathbf{M}(s,0)\mathbf{B}\mathbf{L}(s)\end{aligned}$$

[1] These equations are of the form $AXB - X + C = 0$ and can therefore be solved in cubic time using the dlyap MATLAB command.

which is a Sylvester matrix equation for $\mathbf{M}(s,0)$. For $j \geq 1$, one finds

$$\begin{aligned}
\mathbf{M}(s,j) &= \mathbf{FW}(s,0,j) + \sum_{i=1}^{\infty} \mathbf{R}^i \mathbf{FW}(s,i,j) \\
&= (\mathbf{FL}(s))^{j+1} + \sum_{i=1}^{\infty} \mathbf{R}^i \mathbf{F} \left(\mathbf{W}(s,i,j-1)\mathbf{F} + \mathbf{W}(s,i-1,j)\mathbf{B}\right) \mathbf{L}(s) \\
&= (\mathbf{FL}(s))^{j+1} + \\
&\quad \underbrace{\sum_{i=1}^{\infty} \mathbf{R}^i \mathbf{FW}(s,i,j-1)}_{\mathbf{M}(s,j-1)-\mathbf{FW}(s,0,j-1)} \mathbf{FL}(s) + \mathbf{R} \underbrace{\sum_{i=1}^{\infty} \mathbf{R}^{i-1} \mathbf{FW}(s,i-1,j)}_{\mathbf{M}(s,j)} \mathbf{BL}(s) \\
&= \mathbf{M}(s,j-1)\mathbf{FL}(s) + \mathbf{RM}(s,j)\mathbf{BL}(s),
\end{aligned}$$

whose closed form solution can be obtained using the *vec* operator, by recalling that $vec(XYZ) = (Z^T \otimes X)vec(Y)$

$$\begin{aligned}
&vec(\mathbf{M}(s,j)) \\
&= \left(\mathbf{L}(s)^T \mathbf{F}^T \otimes \mathbf{I}\right) vec(\mathbf{M}(s,j-1)) + \left(\mathbf{L}(s)^T \mathbf{B}^T \otimes \mathbf{R}\right) vec(\mathbf{M}(s,j)) \\
&= \left(\mathbf{I} - \left(\mathbf{L}(s)^T \mathbf{B}^T \otimes \mathbf{R}\right)\right)^{-1} \left(\mathbf{L}(s)^T \mathbf{F}^T \otimes \mathbf{I}\right) vec(\mathbf{M}(s,j-1)) \\
&= \left[\left(\mathbf{I} - \left(\mathbf{L}(s)^T \mathbf{B}^T \otimes \mathbf{R}\right)\right)^{-1} \left(\mathbf{L}(s)^T \mathbf{F}^T \otimes \mathbf{I}\right) \right]^j vec(\mathbf{M}(s,0)). \\
&= \mathbf{U}(s)^j vec(\mathbf{M}(s,0)).
\end{aligned}$$

This implies that $r_2(s)$ can be written in the following manner by using the equality $vec(XYZ) = (Z^T \otimes X)vec(Y)$ with $X = \pi_k$, $Y = \mathbf{M}(s,j)$ and $Z = \mathbf{B}'\left(\mathbf{V}(s,0)\bar{\mathbf{B}}\right)^j \mathbf{S_k}(s)$

$$\begin{aligned}
r_2(s) &= \sum_{j=0}^{\infty} \frac{1}{\lambda} \pi_k \mathbf{M}(s,j) \mathbf{B}' \left(\mathbf{V}(s,0)\bar{\mathbf{B}}\right)^j \mathbf{S_k}(s) \\
&= \frac{1}{\lambda} \sum_{j=0}^{\infty} \left(\mathbf{S_k}(s)^T \left(\bar{\mathbf{B}}^T \mathbf{V}(s,0)^T\right)^j \mathbf{B}'^T \otimes \pi_k\right) vec(\mathbf{M}(s,j)) \\
&= \frac{1}{\lambda} \left(\mathbf{S_k}(s)^T \otimes \pi_k\right) \cdot \\
&\quad \underbrace{\sum_{j=0}^{\infty} \left(\bar{\mathbf{B}}^T \mathbf{V}(s,0)^T \otimes \mathbf{I}\right)^j \left(\mathbf{B}'^T \otimes \mathbf{I}\right) \mathbf{U}(s)^j}_{\mathbf{N}(s)} vec(\mathbf{M}(s,0)) \\
&= \frac{1}{\lambda} \left(\mathbf{S_k}(s)^T \otimes \pi_k\right) \mathbf{N}(s) vec(\mathbf{M}(s,0)),
\end{aligned}$$

where for $\mathbf{N}(s)$ we have

$$\begin{aligned}
\mathbf{N}(s) &= \left(\mathbf{B}'^T \otimes \mathbf{I}\right) + \sum_{j=1}^{\infty} \left(\bar{\mathbf{B}}^T \mathbf{V}(s,0)^T \otimes \mathbf{I}\right)^j \left(\mathbf{B}'^T \otimes \mathbf{I}\right) \mathbf{U}(s)^j \\
&= \left(\mathbf{B}'^T \otimes \mathbf{I}\right) + \left(\bar{\mathbf{B}}^T \mathbf{V}(s,0)^T \otimes \mathbf{I}\right) \mathbf{N}(s)\mathbf{U}(s),
\end{aligned}$$

which completes the proof. □

Theorem 5.8.

$$r_3(s) = \frac{1}{\lambda} \left(\hat{\mathbf{S}}(s)^T \otimes \pi_k\right) \mathbf{N}(s)\mathbf{U}(s)(\mathbf{I} - \mathbf{U}(s))^{-1} vec(\mathbf{M}(s,0)),$$

where $\hat{\mathbf{S}}(s) = \mathbf{V}(s,0)\left(\mathbf{I} - \bar{\mathbf{R}}(s)\right)^{-1}\hat{\mathbf{B}}\mathbb{1}$ *and* $\mathbf{N}(s)$, $\mathbf{U}(s)$ *and* $\mathbf{M}(s,0)$ *are defined in Theorem 5.7.*

Proof. Using $\hat{\mathbf{S}}(s) = \mathbf{V}(s,0)\left(\mathbf{I} - \bar{\mathbf{R}}(s)\right)^{-1}\hat{\mathbf{B}}\mathbb{1}$, we write

$$\begin{aligned}
r_3(s) &= \frac{1}{\lambda} \sum_{\ell=0}^{\infty} \sum_{j=\ell+1}^{\infty} \pi_k \mathbf{M}(s,j) \mathbf{B}' \left(\mathbf{V}(s,0)\bar{\mathbf{B}}\right)^\ell \hat{\mathbf{S}}(s) \\
&= \frac{1}{\lambda} \sum_{\ell=0}^{\infty} \left(\hat{\mathbf{S}}(s)^T \left(\bar{\mathbf{B}}^T \mathbf{V}(s,0)^T\right)^\ell \mathbf{B}'^T \otimes \pi_k\right) \sum_{j=\ell+1}^{\infty} \mathbf{U}(s)^j vec(\mathbf{M}(s,0)) \\
&= \frac{1}{\lambda} \sum_{\ell=0}^{\infty} \left(\hat{\mathbf{S}}(s)^T \left(\bar{\mathbf{B}}^T \mathbf{V}(s,0)^T\right)^\ell \mathbf{B}'^T \otimes \pi_k\right) \cdot \\
&\quad \mathbf{U}(s)^{\ell+1} (\mathbf{I} - \mathbf{U}(s))^{-1} vec(\mathbf{M}(s,0)) \\
&= \frac{1}{\lambda} \left(\hat{\mathbf{S}}(s)^T \otimes \pi_k\right) \underbrace{\sum_{\ell=0}^{\infty} \left(\bar{\mathbf{B}}^T \mathbf{V}(s,0)^T \otimes \mathbf{I}\right)^\ell \left(\mathbf{B}'^T \otimes \mathbf{I}\right) \mathbf{U}(s)^\ell}_{\mathbf{N}(s)} \cdot \\
&\quad \mathbf{U}(s)(\mathbf{I} - \mathbf{U}(s))^{-1} vec(\mathbf{M}(s,0)) \\
&= \frac{1}{\lambda} \left(\hat{\mathbf{S}}(s)^T \otimes \pi_k\right) \mathbf{N}(s)\mathbf{U}(s)(\mathbf{I} - \mathbf{U}(s))^{-1} vec(\mathbf{M}(s,0)),
\end{aligned}$$

□

5.2.2 Computing the response time by spectral expansion. We start with the next theorem which assumes that $\mathbf{V}(s,0)\bar{\mathbf{B}}$ is diagonalizable. For all the numerical experiments performed in this paper, it turns out that these matrices are indeed diagonalizable. As such all the numerical results presented in this paper made use of Theorem 5.9 as it avoids the need to perform a Kronecker expansion.

Theorem 5.9. *When* $\mathbf{V}(s,0)\bar{\mathbf{B}}$ *is diagonalizable for a fixed* s *and its spectral decomposition is* $\sum_{n=1}^{N} \theta_n \mathbf{u}_n \mathbf{v}_n$*, then* $r_2(s)$ *and* $r_3(s)$ *can be computed as*

$$r_2(s) = \frac{1}{\lambda} \pi_k \sum_{n=1}^{N} \tilde{\mathbf{M}}(s,\theta_n) \mathbf{B}' \mathbf{u}_n \mathbf{v}_n \mathbf{S_k}(s),$$

and

$$\begin{aligned}
r_3(s) = &\frac{1}{\lambda} \pi_k \sum_{n=1}^{N} \frac{\tilde{\mathbf{M}}(s,1) - \tilde{\mathbf{M}}(s,\theta_n)}{1-\theta_n} \mathbf{B}' \mathbf{u}_n \mathbf{v}_n \\
&\cdot \mathbf{V}(s,0)\left(\mathbf{I} - \bar{\mathbf{R}}(s)\right)^{-1} \hat{\mathbf{B}}\mathbb{1},
\end{aligned}$$

where $\tilde{\mathbf{M}}(s,\theta)$ *is the solution of the Sylvester equation*[2]

$$\tilde{\mathbf{M}}(s,\theta) = \mathbf{FL}(s) + \theta\tilde{\mathbf{M}}(s,\theta)\mathbf{FL}(s) + \mathbf{R}\tilde{\mathbf{M}}(s,\theta)\mathbf{BL}(s). \tag{23}$$

Proof. Using $\mathbf{M}(s,j)$ defined in Theorem 5.7, $r_2(s)$ and $r_3(s)$ can be written as

$$r_2(s) = \sum_{j=0}^{\infty} \frac{1}{\lambda} \pi_k \mathbf{M}(s,j) \mathbf{B}' \left(\mathbf{V}(s,0)\bar{\mathbf{B}}\right)^j \mathbf{S_k}(s),$$

$$r_3(s) = \sum_{j=1}^{\infty} \frac{1}{\lambda} \pi_k \mathbf{M}(s,j) \mathbf{B}' \sum_{\ell=0}^{j-1} \left(\mathbf{V}(s,0)\bar{\mathbf{B}}\right)^\ell \mathbf{V}(s,0)\left(\mathbf{I} - \bar{\mathbf{R}}(s)\right)^{-1} \hat{\mathbf{B}}\mathbb{1},$$

where $\mathbf{M}(s,j)$ was shown to satisfy

$$\begin{aligned}
\mathbf{M}(s,0) &= \mathbf{FL}(s) + \mathbf{RM}(s,0)\mathbf{BL}(s), \\
\mathbf{M}(s,j) &= \mathbf{M}(s,j-1)\mathbf{FL}(s) + \mathbf{RM}(s,j)\mathbf{BL}(s),
\end{aligned}$$

[2] This equation is of the form $AXB + XC = D$ and can be solved in cubic time using a Hessenberg decomposition of (B, C) and Schur decomposition of A, see [4].

for $j \geq 1$. Multiplying the last equation with θ^j and summing over j gives

$$\sum_{j=1}^{\infty} \theta^j \mathbf{M}(s,j) = \sum_{j=1}^{\infty} \theta^j \mathbf{M}(s,j-1)\mathbf{F}\mathbf{L}(s) + \mathbf{R}\sum_{j=1}^{\infty} \theta^j \mathbf{M}(s,j)\mathbf{B}\mathbf{L}(s).$$

For $\tilde{\mathbf{M}}(s,\theta) = \sum_{j=0}^{\infty} \theta^j \mathbf{M}(s,j)$ we therefore have

$$\tilde{\mathbf{M}}(s,\theta) - \mathbf{M}(s,0) = \theta\tilde{\mathbf{M}}(s,\theta)\mathbf{F}\mathbf{L}(s) + \mathbf{R}\tilde{\mathbf{M}}(s,\theta)\mathbf{B}\mathbf{L}(s) - \mathbf{R}\mathbf{M}(s,0)\mathbf{B}\mathbf{L}(s),$$

from which, for any s and θ, $\tilde{\mathbf{M}}(s,\theta)$ is the solution of the Sylvester equation

$$\tilde{\mathbf{M}}(s,\theta) = \mathbf{F}\mathbf{L}(s) + \theta\tilde{\mathbf{M}}(s,\theta)\mathbf{F}\mathbf{L}(s) + \mathbf{R}\tilde{\mathbf{M}}(s,\theta)\mathbf{B}\mathbf{L}(s). \tag{24}$$

If $\mathbf{V}(s,0)\bar{\mathbf{B}}$ is diagonalizable and its spectral decomposition is $\sum_{n=1}^{N} \theta_n \mathbf{u}_n\mathbf{v}_n$ then

$$\begin{aligned} r_2(s) &= \sum_{j=0}^{\infty} \frac{1}{\lambda}\pi_k \mathbf{M}(s,j)\mathbf{B}' \sum_{n=1}^{N} \theta_n^j \mathbf{u}_n\mathbf{v}_n\mathbf{S}_{\mathbf{k}}(s) \\ &= \frac{1}{\lambda}\pi_k \sum_{n=1}^{N} \tilde{\mathbf{M}}(s,\theta_n)\mathbf{B}'\mathbf{u}_n\mathbf{v}_n\mathbf{S}_{\mathbf{k}}(s), \end{aligned}$$

and

$$\begin{aligned} r_3(s) &= \sum_{j=1}^{\infty} \frac{1}{\lambda}\pi_k \mathbf{M}(s,j)\mathbf{B}' \sum_{\ell=0}^{j-1}\sum_{n=1}^{N} \theta_n^\ell \mathbf{u}_n\mathbf{v}_n\mathbf{V}(s,0)\big(\mathbf{I} - \bar{\mathbf{R}}(s)\big)^{-1}\hat{\mathbf{B}}\mathbb{1} \\ &= \frac{1}{\lambda}\pi_k \sum_{n=1}^{N}\sum_{j=1}^{\infty} \mathbf{M}(s,j)\mathbf{B}' \frac{1-\theta_n^j}{1-\theta_n}\mathbf{u}_n\mathbf{v}_n\mathbf{V}(s,0)\big(\mathbf{I} - \bar{\mathbf{R}}(s)\big)^{-1}\hat{\mathbf{B}}\mathbb{1} \\ &= \frac{1}{\lambda}\pi_k \sum_{n=1}^{N} \frac{\tilde{\mathbf{M}}(s,1) - \tilde{\mathbf{M}}(s,\theta_n)}{1-\theta_n}\mathbf{B}'\mathbf{u}_n\mathbf{v}_n\mathbf{V}(s,0)\big(\mathbf{I} - \bar{\mathbf{R}}(s)\big)^{-1}\hat{\mathbf{B}}\mathbb{1}. \end{aligned}$$

□

If $\mathbf{V}(s,0)\bar{\mathbf{B}}$ is non-diagonalizable, then its spectral decomposition contains at least one non-trivial Jordan block. The following theorem discusses the case when the spectral decomposition of $\mathbf{V}(s,0)\bar{\mathbf{B}}$ is composed by a single Jordan block of maximal size.

Theorem 5.10. *If for a fixed s the spectral decomposition of $\mathbf{V}(s,0)\bar{\mathbf{B}}$ is*

$$\mathbf{V}(s,0)\bar{\mathbf{B}} = \mathbf{\Theta}^{-1}\begin{bmatrix} \theta & 1 & & \\ & \theta & 1 & \\ & & \ddots & \ddots \\ & & & \theta \end{bmatrix}\mathbf{\Theta},$$

then

$$\begin{aligned} r_2(s) &= \frac{1}{\lambda}\pi_k \mathbf{\Phi}(s)\mathbf{\Theta}\mathbf{S}_{\mathbf{k}}(s), \\ r_3(s) &= \frac{1}{\lambda}\pi_k \mathbf{\Psi}(s)\mathbf{\Theta}\mathbf{V}(s,0)\big(\mathbf{I} - \bar{\mathbf{R}}(s)\big)^{-1}\hat{\mathbf{B}}\mathbb{1}, \end{aligned}$$

where the ℓth column of $\mathbf{\Phi}(s)$ and $\mathbf{\Psi}(s)$ can be computed as

$$[\mathbf{\Phi}(s)]_\ell = \sum_{n=0}^{\ell-1} \frac{\tilde{\mathbf{M}}^{(n)}(s,\theta)}{n!}[\mathbf{B}'\mathbf{\Theta}^{-1}]_{\ell-n},$$

$$[\mathbf{\Psi}(s)]_\ell = \sum_{n=0}^{\ell-1}\left(\frac{\tilde{\mathbf{M}}(s,1)}{(1-\theta)^{n+1}} - \sum_{u=0}^{n} \frac{1}{(1-\theta)^{u+1}}\frac{\tilde{\mathbf{M}}^{(n-u)}(s,\theta)}{(n-u)!}\right)[\mathbf{B}'\mathbf{\Theta}^{-1}]_{\ell-n},$$

such that $\tilde{\mathbf{M}}^{(0)}(s,\theta) = \tilde{\mathbf{M}}(s,\theta)$ and $\tilde{\mathbf{M}}^{(n)}(s,\theta)$ ($n \geq 1$) is the solution of the Sylvester equation[3]

$$\begin{aligned} \tilde{\mathbf{M}}^{(n)}(s,\theta) = n\tilde{\mathbf{M}}^{(n-1)}(s,\theta)\mathbf{F}\mathbf{L}(s) \\ + \theta\tilde{\mathbf{M}}^{(n)}(s,\theta)\mathbf{F}\mathbf{L}(s) + \mathbf{R}\tilde{\mathbf{M}}^{(n)}(s,\theta)\mathbf{B}\mathbf{L}(s). \end{aligned} \tag{25}$$

Proof. First we note that (25) is the nth derivative of (24) with respect to θ. (25) is a Sylvester equation for $\tilde{\mathbf{M}}^{(n)}(s,\theta)$ when $\tilde{\mathbf{M}}^{(n-1)}(s,\theta)$ is known. It means that for a fixed θ, $\tilde{\mathbf{M}}^{(n)}(s,\theta)$ can be computed iteratively starting from $n = 0$.

For the given spectral decomposition of $\mathbf{V}(s,0)\bar{\mathbf{B}}$, we have

$$\begin{aligned} r_2(s) &= \sum_{j=0}^{\infty} \frac{1}{\lambda}\pi_k \mathbf{M}(s,j)\mathbf{B}'\mathbf{\Theta}^{-1}\begin{bmatrix} \theta & 1 & & \\ & \theta & 1 & \\ & & \ddots & \ddots \\ & & & \theta \end{bmatrix}^j \mathbf{\Theta}\mathbf{S}_{\mathbf{k}}(s) \\ &= \underbrace{\frac{1}{\lambda}\pi_k \sum_{j=0}^{\infty} \mathbf{M}(s,j)\mathbf{B}'\mathbf{\Theta}^{-1}\begin{bmatrix} \theta^j & \binom{j}{1}\theta^{j-1} & \dots & \binom{j}{N}\theta^{j-N} \\ & \theta^j & \binom{j}{1}\theta^{j-1} & \dots \\ & & \ddots & \ddots \\ & & & \theta^j \end{bmatrix}}_{\mathbf{\Phi}(s)} \mathbf{\Theta}\mathbf{S}_{\mathbf{k}}(s), \end{aligned}$$

where $\binom{j}{i} = 0$ for $i > j$.

The first column of $\mathbf{\Phi}(s)$, $[\mathbf{\Phi}(s)]_1$, can be computed as

$$[\mathbf{\Phi}(s)]_1 = \sum_{j=0}^{\infty} \mathbf{M}(s,j)[\mathbf{B}'\mathbf{\Theta}^{-1}]_1\theta^j = \tilde{\mathbf{M}}(s,\theta)[\mathbf{B}'\mathbf{\Theta}^{-1}]_1.$$

For the second column we have

$$\begin{aligned} [\mathbf{\Phi}(s)]_2 &= \sum_{j=1}^{\infty} \mathbf{M}(s,j)[\mathbf{B}'\mathbf{\Theta}^{-1}]_1\binom{j}{1}\theta^{j-1} + \sum_{j=0}^{\infty} \mathbf{M}(s,j)[\mathbf{B}'\mathbf{\Theta}^{-1}]_2\theta^j \\ &= \tilde{\mathbf{M}}^{(1)}(s,\theta)[\mathbf{B}'\mathbf{\Theta}^{-1}]_1 + \tilde{\mathbf{M}}(s,\theta)[\mathbf{B}'\mathbf{\Theta}^{-1}]_2. \end{aligned}$$

and the last column is

$$\begin{aligned} [\mathbf{\Phi}(s)]_N &= \sum_{n=0}^{N-1}\sum_{j=n}^{\infty} \mathbf{M}(s,j)[\mathbf{B}'\mathbf{\Theta}^{-1}]_{N-n}\binom{j}{n}\theta^{j-n} \\ &= \sum_{n=0}^{N-1} \frac{\tilde{\mathbf{M}}^{(n)}(s,\theta)}{n!}[\mathbf{B}'\mathbf{\Theta}^{-1}]_{N-n}. \end{aligned}$$

Similarly,

$$r_3(s) = \frac{1}{\lambda}\pi_k \mathbf{\Psi}(s)\mathbf{\Theta}\mathbf{V}(s,0)\big(\mathbf{I} - \bar{\mathbf{R}}(s)\big)^{-1}\hat{\mathbf{B}}\mathbb{1},$$

[3] This equation is of the same form as (23) and can therefore also be solved in cubic time.

SCV	p	$1/\gamma_1$	$1/\gamma_2$
1	0.5000	1	1
2	0.7887	0.6340	2.3660
5	0.9082	0.5505	5.4495
10	0.9523	0.5251	10.4749
19	0.9743	0.5132	19.4868

Table 1: Parameter settings of the hyper-exponential job size distribution for various SCV values.

where

$$\mathbf{\Psi}(s) = \sum_{j=1}^{\infty} \mathbf{M}(s,j)\mathbf{B}'\mathbf{\Theta}^{-1} \sum_{\ell=0}^{j-1} \begin{bmatrix} \theta^\ell & \binom{\ell}{1}\theta^{\ell-1} & \cdots & \binom{\ell}{N}\theta^{\ell-N} \\ & \theta^\ell & \binom{\ell}{1}\theta^{\ell-1} & \cdots \\ & & \ddots & \ddots \\ & & & \theta^\ell \end{bmatrix}$$

$$= \sum_{j=1}^{\infty} \mathbf{M}(s,j)\mathbf{B}'\mathbf{\Theta}^{-1} \begin{bmatrix} \frac{1-\theta^j}{1-\theta} & \frac{d}{d\theta}\frac{1-\theta^j}{1-\theta} & \cdots & \frac{1}{N!}\frac{d^N}{d\theta^N}\frac{1-\theta^j}{1-\theta} \\ & \frac{1-\theta^j}{1-\theta} & \frac{d}{d\theta}\frac{1-\theta^j}{1-\theta} & \cdots \\ & & \ddots & \ddots \\ & & & \frac{1-\theta^j}{1-\theta} \end{bmatrix}.$$

The first column of $\mathbf{\Psi}(s)$ is

$$[\mathbf{\Psi}(s)]_1 = \sum_{j=0}^{\infty} \mathbf{M}(s,j)[\mathbf{B}'\mathbf{\Theta}^{-1}]_1 \frac{1-\theta^j}{1-\theta} = \frac{\tilde{\mathbf{M}}(s,1) - \tilde{\mathbf{M}}(s,\theta)}{1-\theta}[\mathbf{B}'\mathbf{\Theta}^{-1}]_1.$$

and its Nth column is

$$[\mathbf{\Psi}(s)]_N = \sum_{n=0}^{N-1} \sum_{j=n}^{\infty} \mathbf{M}(s,j)[\mathbf{B}'\mathbf{\Theta}^{-1}]_{N-n} \frac{1}{n!}\frac{d^n}{d\theta^n}\frac{1-\theta^j}{1-\theta}$$

$$= \sum_{n=0}^{N-1} \frac{1}{n!}\frac{d^n}{d\theta^n}\frac{\tilde{\mathbf{M}}(s,1) - \tilde{\mathbf{M}}(s,\theta)}{1-\theta}[\mathbf{B}'\mathbf{\Theta}^{-1}]_{N-n},$$

where

$$\frac{d^n}{d\theta^n}\frac{\tilde{\mathbf{M}}(s,1) - \tilde{\mathbf{M}}(s,\theta)}{1-\theta} = \frac{n!\ \tilde{\mathbf{M}}(s,1)}{(1-\theta)^{n+1}} - \sum_{u=0}^{n} \frac{n!}{(n-u)!}\frac{\tilde{\mathbf{M}}^{(n-u)}(s,\theta)}{(1-\theta)^{u+1}}$$

□

The cases when multiple, potentially non-trivial Jordan blocks appear in the spectral decomposition of $\mathbf{V}(s,0)\bar{\mathbf{B}}$ can be handled by the combination of these results and are omitted here.

6 NUMERICAL RESULTS

In this section we present numerical results obtained by numerically inverting the Laplace transform $r(s)$ of the response time distribution. As we are mainly interested in systems where the workload consists of a mixture of long and short jobs, we make use of a hyper-exponential (HEXP) distribution with $n_s = 2$ phases to model the job size distribution. Under a 2-phase HEXP distribution a job has an exponentially distributed length with mean $1/\gamma_1$ with probability p and an exponentially distributed length with mean $1/\gamma_2$ with probability $1-p$. The parameters p, γ_1 and γ_2 are set by matching the mean job duration EX, its squared coefficient of variation SCV (for any $SCV \geq 1$) and a shape parameter. Unless otherwise stated, we set

$$EX = 1,\ \ \gamma_1 = 1 + \sqrt{\frac{SCV-1}{SCV+1}},\ \ \gamma_2 = 1 - \sqrt{\frac{SCV-1}{SCV+1}},\ \ p = \frac{\gamma_1}{2},$$

and

$$\alpha = \begin{bmatrix} p & 1-p \end{bmatrix},\ \ \mathbf{A} = \begin{bmatrix} -\gamma_1 & \\ & -\gamma_2 \end{bmatrix}.$$

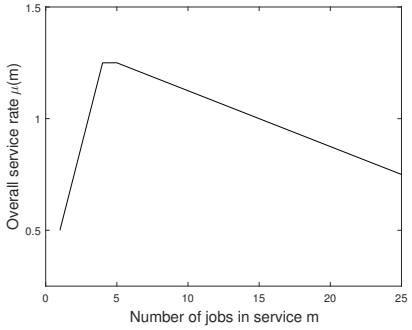

Figure 1: Prototypical service rate curve of [5].

Table 1 lists the resulting parameters for $SCV = 1, 2, 5, 10$ and 19. For instance, when $SCV = 10$ about 95% of the jobs are type 1 jobs, these are typically *short* jobs, and the remaining 5% of the jobs have a job length that is on average 20 times longer. We make use of the same prototypical service rate curve $\mu(m)$ as in [5] which is depicted in Figure 1. Note that the service rate is maximized at when $k = 4$ or 5.

To numerically invert the Laplace transform we make use of the Euler algorithm [1, Section 5]. To determine the probability $P[R \leq t]$ that the response time is at most t, this algorithm evaluates $r(s)/s$ in a set of $2M$ points, where $Re(s) = M\ln(10)/3t > 0$. For our numerical results we set $M = 32$ and all computations were performed in double precision using the MATLAB function euler_inversion which can be downloaded from the Mathworks File Exchange server [8]. This function requires the function that evaluates $r(s)/s$ for a specific value of s as input.

As stated in Section 4 the theorems presented in this paper make use of matrices of size $n_s^k n_a$, which allows us to specify the matrices in an elegant manner using Kronecker sums and products. Even if $n_s = 2$ this would result in matrices of size $2^k n_a$. To reduce the computation times, we use the common approach to reduce the size of the matrices involved by keeping track of the number of jobs in each phase (instead of the service phase of each individual job), e.g., [14]. For $n_s = 2$ this would reduce the size of the matrices to $(k+1)n_a$ (as there can be between 0 and k jobs in phase 1). However, looking at the analysis in Section 5 it should be noted that we always need to maintain the phase of the tagged job (located in server 1). This implies that we can only collapse the phases of the remaining $k-1$ service phases and the size of the matrices used in our computations therefore equals $2kn_a$.

For all the numerical experiments presented in this section we were able to make use of Theorem 5.9 as the matrices $\mathbf{V}(s,0)\bar{\mathbf{B}}$ turned out to be diagonalizable for all the parameter settings considered. For Poisson input ($\lambda = 0.8$) and HEXP service the computation

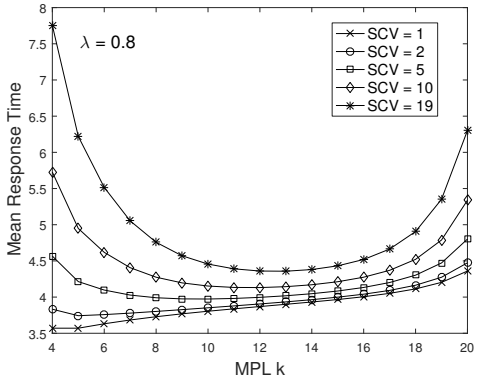

Figure 2: Mean job response time as a function of the MPL k under Poisson arrivals with rate $\lambda = 0.8$.

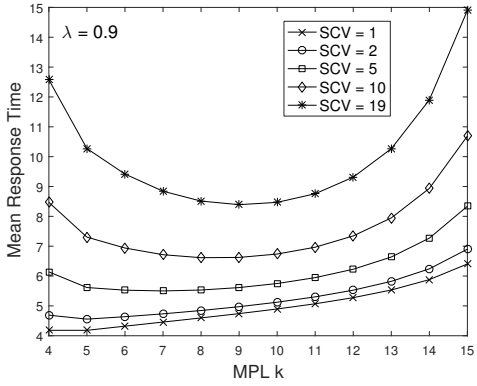

Figure 3: Mean job response time as a function of the MPL k under Poisson arrivals with rate $\lambda = 0.9$.

time was less than 1 second for $k = 6$, about 3.2 seconds for $k = 10$ and 7.5 seconds for $k = 15$ to compute a single probability of the form $P[R > t]$ (on an Intel Core i7-3630QM 2.40 GHz laptop). These computation times are not very sensitive to the specific value of t or the SCV of the job sizes. For MAP input with $n_a = 2$ instead of Poisson input the computation times roughly increase by a factor 8 (as the size of the matrices doubles and computation times are cubic in the size of the matrices involved). If we rely on Theorems 5.7 and 5.8 instead of Theorem 5.9 the computation times for $k = 6, 10$ and 15 increase to approximately 4, 40 and 440 seconds, which clearly demonstrates the effectiveness of Theorem 5.9 to reduce the computation times.

6.1 Poisson arrivals

The Poisson arrival process with rate λ is a special case of a MAP with $n_a = 1$, $\mathbf{D_0} = -\lambda$ and $\mathbf{D_1} = \lambda$. Figures 2 and 3 depict the mean response time in case of Poisson job arrivals with rate $\lambda = 0.8$ and 0.9 for various choices of the service time SCV. These plots are similar to [5, Figure 3] and confirm that selecting a higher MPL k when the job sizes are highly variable is beneficial for the mean response time.

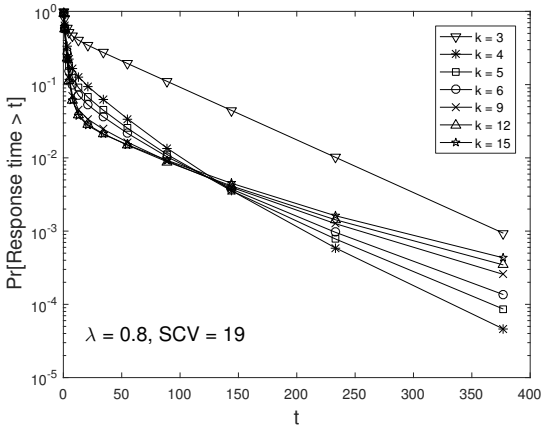

Figure 4: Response time distribution for various MPL k values under Poisson arrivals with rate $\lambda = 0.8$ and $SCV = 19$.

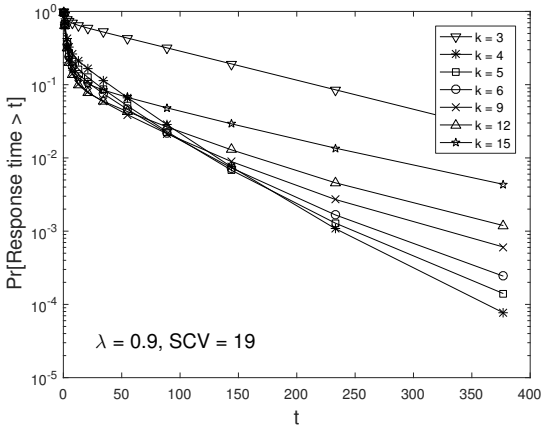

Figure 5: Response time distribution for various MPL k values under Poisson arrivals with rate $\lambda = 0.9$ and $SCV = 19$.

Figures 4 and 5 show the response time distribution for various choices of the MPL k for the setting with $SCV = 19$. These figures clearly illustrate that while the mean response time is minimized for an MPL value well beyond 5 (where the service rate curve is maximized), the tail of the response time distribution behaves very differently and is minimized by setting $k = 4$, the smallest k value for which $\mu(k) \geq \mu(m)$ for all m. This observation is in agreement with the observation made in [9] that under light tailed job size distributions, the tail asymptotics of a limited processor sharing queue (with a fixed service rate curve) tends to improve as the MPL level k decreases. It also confirms our intuition that the mean response time improves up to some point as for larger k short jobs have an easier time passing long jobs, but this makes it harder for the long jobs to complete service, which worsens the tail behavior of the response time distribution.

This is further illustrated in Figures 6 and 7 where we depict the response time distribution conditioned on the initial service phase of a job. We state that jobs that started service in phase i, for $i = 1, 2$, are type-i jobs. In case of HEXP job lengths, this implies that the

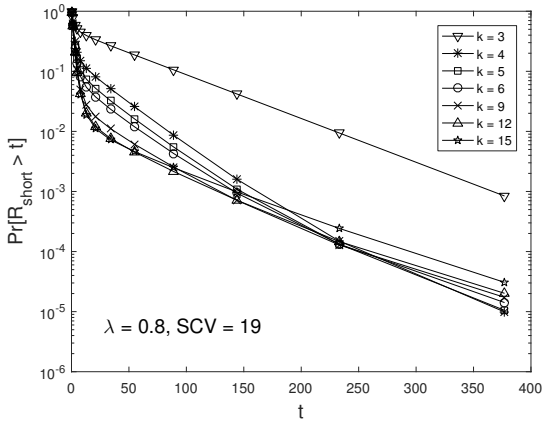

Figure 6: Response time distribution of *short* jobs, i.e., type-1 jobs, for various MPL k values under Poisson arrivals with rate $\lambda = 0.8$ and $SCV = 19$.

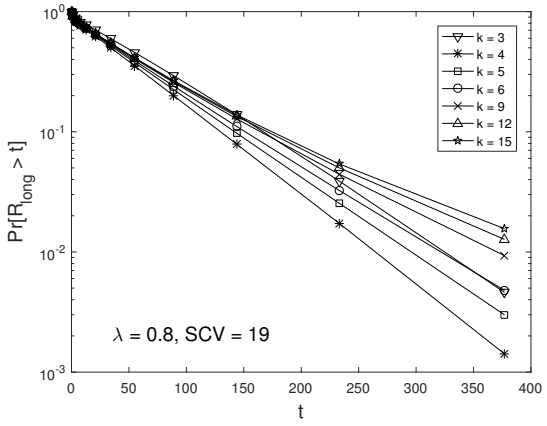

Figure 7: Response time distribution of *long* jobs, i.e., type-2 jobs, for various MPL k values under Poisson arrivals with rate $\lambda = 0.8$ and $SCV = 19$.

type-i jobs have a mean length of $1/\gamma_i$ and roughly speaking the type-1 jobs correspond to short jobs (with a mean length of about 0.5) and type-2 jobs are typically long (with a mean close to 20). To compute these distributions, denoted as R_{short} and R_{long}, we can rely on the results presented in Section 5, except that the vector α used in the matrices $\mathbf{B'}$ and $\mathbf{F'}$ needs to be replaced by e_i, the stochastic vector with entry i equal to 1. Ideally we would like to compute the response time distribution of a job given its size, but this distribution cannot be directly obtained from Section 5.

Figure 6 shows that the response time distribution of the *short* jobs tends to improve as k increases up to $k = 12$, the value that minimizes the mean response time. For larger t smaller k values still perform better, but this is most likely due to the fact that not all type-1 jobs are truly short. Similarly Figure 7 shows that increasing k is typically bad for the *long* jobs.

We end this section by depicting the 99^{th} percentile of the response time distribution for various choices of the *SCV* in Figure 8.

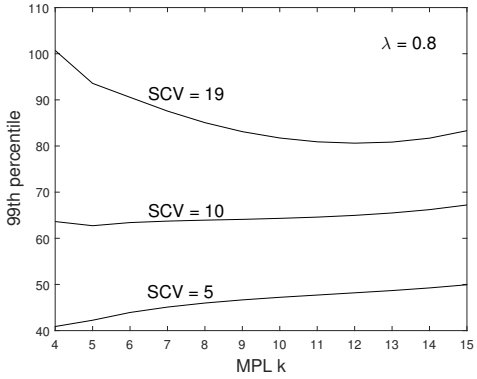

Figure 8: The 99^{th} percentile of the response time distribution as a function of the MPL k under Poisson arrivals with $\lambda = 0.8$.

This percentile is very meaningful whenever the main concern of the system is to guarantee good response time for the majority of the jobs, e.g., 99% of the jobs. Figure 8 indicates that except for the $SCV = 19$ case, the 99^{th} percentile is minimized by setting the MPL equal to 4 or 5. It indicates that the tail asymptotics kick in rather quickly and using a larger MPL k might not give the desirable result. The fact that a larger k reduces the 99^{th} percentile of the response time for an *SCV* equal to 19 is not surprising as in this case about 97.5% of the jobs are type-1 jobs (see Table 1) and these benefit from increased k values.

6.2 Phase-type renewal arrivals

In this section we investigate the impact of the arrival process by replacing the Poisson arrivals used in the previous section with a phase-type renewal process. This means that consecutive inter-arrival times are still independent, but the inter-arrival time distribution follows a phase-type distribution. Similar to the service time distribution, we assume an HEXP inter-arrival time distribution characterized by the mean arrival rate λ and its squared coefficient of variation SCV_a, that is

$$n_a = 2, \quad \mathbf{D_0} = \lambda \begin{bmatrix} -\gamma_1 & \\ & -\gamma_2 \end{bmatrix}, \quad \mathbf{D_1} = \lambda \begin{bmatrix} p\gamma_1 & (1-p)\gamma_1 \\ p\gamma_2 & (1-p)\gamma_2 \end{bmatrix},$$

where $\gamma_{1,2} = 1 \pm \sqrt{(SCV_a - 1)/(SCV_a + 1)}$ and $p = \gamma_1/2$.

Figure 9 depicts the mean response time as a function of k when the PH renewal process has a mean arrival rate $\lambda = 0.8$ and an $SCV_a = 10$. Note in such case the inter-arrival times are a mixture of many rather short inter-arrival times and less frequent long inter-arrival times. The most notable result in this figure (which we also confirmed by simulation) is that when the MPL k is large, the mean response time reduces as the job sizes become more variable (i.e., *SCV* increases). The intuition behind this rather unexpected result is as follows. In case of bursty arrivals there are some time intervals where the queue received many jobs in a short period of time. When these jobs are a mixture of long and short jobs and k is large, some of the jobs, being the short ones, can be cleared more quickly. This implies that the queue length can be reduced more quickly after such a burst of arrivals in case the job sizes are

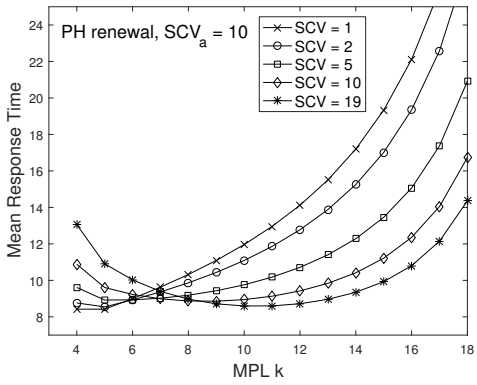

Figure 9: Mean job response time as a function of the MPL k under PH renewal input with rate $\lambda = 0.8$ and $SCV_a = 10$ for various SCV values for the job sizes.

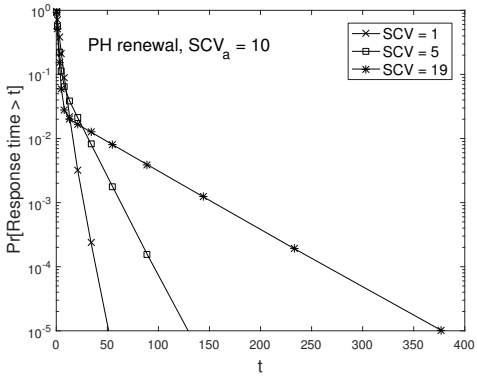

Figure 10: Response time distribution for $k = 15$ under PH renewal input with rate $\lambda = 0.8$ and $SCV_a = 10$ for job sizes with $SCV = 1, 5$ and 19.

more variable. If the queue length decreases more quickly, the total service capacity increases more rapidly after such a burst of arrivals, which is clearly beneficial for the response times. Another example that confirms this intuition is presented further on in case of MAP arrivals with correlated inter-arrival times. Figure 10 indicates that while the mean response times may improve with the variability of the job sizes when k is large, e.g., $k = 15$, the tail probabilities behave in line with our expectations: more variable job sizes given higher tail probabilities.

Figures 9 and 11 further illustrate that increasing the MPL k beyond k^* also reduces the mean response time in case of PH renewal input and the tail probabilities are affected in a similar manner as in the Poisson case (that is, setting $k = k^*$ remains optimal for the tail).

6.3 Markov modulated Poisson arrivals

In this section we look at the impact of having correlated job inter-arrival times. For this propose we rely on a 2-state Markov modulated Poisson process (MMPP). Such a process alternates between two states and while in state i it generates arrivals according to

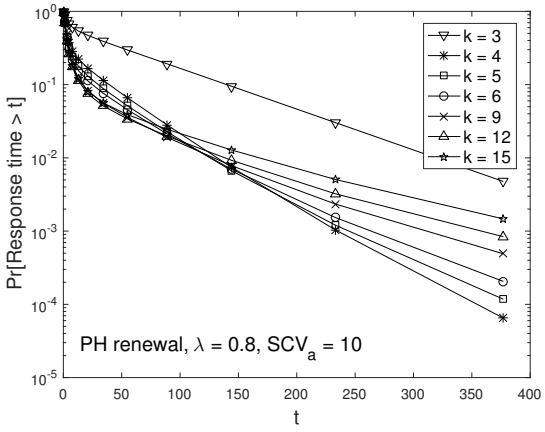

Figure 11: Response time distribution for various MPL k values under PH renewal input with rate $\lambda = 0.8$ and $SCV_a = 10$ and job sizes with $SCV = 19$.

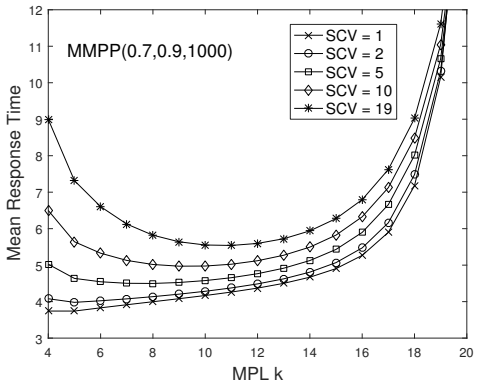

Figure 12: Mean job response time as a function of the MPL k under MMPP input with rates $\lambda_1 = 0.7$, $\lambda_2 = 0.9$ and a mean sojourn time $1/\phi = 1000$.

a Poisson process with rate λ_i, for $i = 1, 2$. The sojourn times in each state are exponential and we assume they have the same mean $1/\phi$. Such a process can be represented as a MAP in the following manner:

$$n_a = 2, \ \mathbf{D_0} = \begin{bmatrix} -\lambda_1 - \phi & \phi \\ \phi & -\lambda_2 - \phi \end{bmatrix}, \ \mathbf{D_1} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}.$$

Figure 12 depicts the mean response time as a function of the MPL k for various SCV values of the job size in case of MMPP input with $\lambda_1 = 0.7$, $\lambda_2 = 0.9$ and $1/\phi = 1000$, while Figure 13 depicts the sojourn time distribution for the same MMPP when $SCV = 19$. The main conclusions are the same as in the Poisson setting: under high job size variability setting the MPL k beyond k^* reduces the mean response time at the expense of an increase in the tail probabilities.

Figure 12 may appear to be in conflict with the intuition provided in the previous section in case the arrivals occur in a bursty manner as even for larger k, a higher SCV value results in a larger mean response time (in contrast to Figure 9). However, this is merely due to the fact that an MMPP process with $\lambda_1 = 0.7$ and $\lambda_2 = 0.9$ is

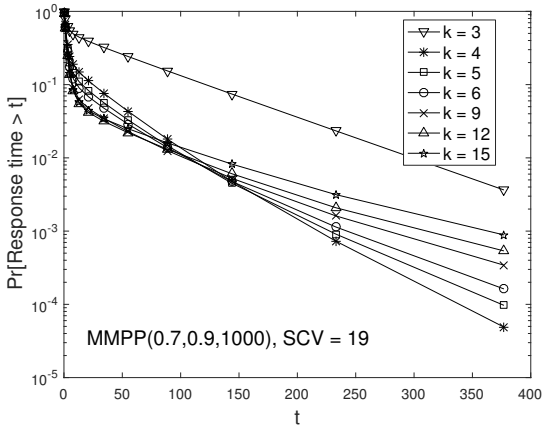

Figure 13: Response time distribution for $k = 15$ under MMPP input with rates $\lambda_1 = 0.7$, $\lambda_2 = 0.9$ and a mean sojourn time $1/\phi = 1000$ for job sizes with $SCV = 19$.

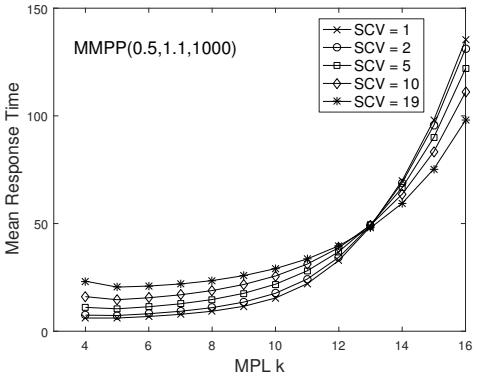

Figure 14: Mean job response time as a function of the MPL k under MMPP input with rates $\lambda_1 = 0.5$, $\lambda_2 = 1.1$ and a mean sojourn time $1/\phi = 1000$.

not very bursty. If we further increase the difference between the arrival rates λ_1 and λ_2 to 0.6 as illustrated in Figure 14 we note that more variable job sizes do result in lower mean response times for k large.

7 CONCLUSIONS

Motivated by time sharing systems, we studied a processor sharing queueing system where at most k jobs are served simultaneously, the overall service rate $\mu(m)$ depends on the number of jobs m in service and additional jobs are buffered (and served in FCFS order). Arrivals are assumed to occur according to a Markovian arrival process (MAP) and the job sizes follow a phase-type (PH) distribution. We derived an expression for the Laplace transform of the response time distribution via both a Kronecker and spectral expansion approach.

By numerically inverting the Laplace transform we demonstrated that while increasing the multi-programming level (MPL) k beyond $k^* = \arg\max_m \mu(m)$ may decrease the mean response time in case of highly variable job sizes (as shown in [5]), this has a negative effect on the tail behavior which may already be visible at the 99th percentile of the response time distribution. Further, in case of bursty arrivals and large MPL k having more variable job sizes may reduce the mean response times, while the tails still behave as expected (more job size variability leads to larger tail probabilities).

In short the results shed light on the potential drawbacks associated with increasing the MPL value beyond k^* when performing admission control.

ACKNOWLEDGMENTS

The authors are grateful to Gábor Horváth and Illés Horváth for some helpful discussions.

REFERENCES

[1] J. Abate and W. Whitt. 2006. A unified framework for numerically inverting Laplace transforms. *INFORMS Journal on Computing* 18, 4 (2006), 408–421.
[2] B. Avi-Itzhak and S. Halfin. 1988. Expected response times in a non-symmetric time sharing queue with a limited number of service positions. In *Proceedings of ITC*, Vol. 12. 2–1.
[3] P. Brémaud. 2013. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues.* Vol. 31. Springer Science & Business Media.
[4] J.D. Gardiner, A.J. Laub, J.J. Amato, and C.B. Moler. 1992. Solution of the Sylvester matrix $AXB^T + CXD^T = E$. *ACM Trans. Math. Software* 18, 2 (1992), 223–231.
[5] V. Gupta and M. Harchol-Balter. 2009. Self-adaptive Admission Control Policies for Resource-sharing Systems. *SIGMETRICS Perform. Eval. Rev.* 37, 1 (June 2009), 311–322. https://doi.org/10.1145/2492101.1555385
[6] J. Kriege and P. Buchholz. 2014. *PH and MAP Fitting with Aggregated Traffic Traces.* Springer International Publishing, Cham, 1–15. https://doi.org/10.1007/978-3-319-05359-2_1
[7] G. Latouche and V. Ramaswami. 1999. *Introduction to Matrix Analytic Methods in Stochastic Modeling.* Society for Industrial and Applied Mathematics, Philadelphia.
[8] T. McClure. 2013. Numerical Inverse Laplace Transform. Computer software. Mathworks File Exchange. (2013).
[9] J. Nair, A. Wierman, and B. Zwart. 2010. Tail-robust Scheduling via Limited Processor Sharing. *Perform. Eval.* 67, 11 (Nov. 2010), 978–995. https://doi.org/10.1016/j.peva.2010.08.012
[10] M.F. Neuts. 1981. *Matrix Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore.
[11] M. Nuyens and W. van der Weij. 2009. Monotonicity in the limited processor-sharing queue. *Stochastic Models* 25, 3 (2009), 408–419.
[12] K.M. Rege and B. Sengupta. 1988. Response time distribution in a multiprogrammed computer with terminal traffic. *Performance Evaluation* 8, 1 (1988), 41–50.
[13] M. Welsh, D. Culler, and E. Brewer. 2001. SEDA: An Architecture for Well-conditioned, Scalable Internet Services. *SIGOPS Oper. Syst. Rev.* 35, 5 (Oct. 2001), 230–243. https://doi.org/10.1145/502059.502057
[14] F. Zhang and L. Lipsky. 2006. Modelling Restricted Processor Sharing.. In *PDPTA*. 353–359.
[15] J. Zhang, JG Dai, and B. Zwart. 2009. Law of large number limits of limited processor-sharing queues. *Mathematics of Operations Research* 34, 4 (2009), 937–970.
[16] J. Zhang, JG Dai, and B. Zwart. 2011. Diffusion limits of limited processor sharing queues. *The Annals of Applied Probability* 21, 2 (2011), 745–799.
[17] J. Zhang and B. Zwart. 2008. Steady state approximations of limited processor sharing queues in heavy traffic. *Queueing Systems* 60, 3 (2008), 227–246.