Randomized Load Balancing in Finite Regimes

Antonie S. Godtschalk
Capgemini

Florin Ciucu
University of Warwick

***Abstract*—Randomized load balancing is a cost efficient policy for job scheduling in parallel server queueing systems whereby, with every incoming job, a central dispatcher randomly polls some servers and selects the one with the smallest queue. By exactly deriving the jobs' delay distribution in such systems, in explicit and closed form, Mitzenmacher [13] proved the so-called 'power-of-two' result, which states that the random polling of only two servers yields an exponential improvement in delay over randomly selecting a single server. Such a fundamental result, however, was obtained in an asymptotic regime in the total number of servers, and does do not necessarily provide accurate estimates for practical finite regimes with small or moderate number of servers. In this paper we obtain stochastic lower and upper bounds on the jobs' average delay in non-asymptotic/finite regimes, by extending ideas for analyzing the particular case of the Join-the-Shortest-Queue (JSQ) policy. Numerical illustrations indicate not only that the (lower) bounds are remarkably accurate, but also that the asymptotic approximation can be misleading in scenarios with a small number of servers, and especially at very high utilizations.**

I. Introduction

Parallel server queueing systems model a wide range of scenarios related to daily situations, e.g., toll booths, bank tellers, supermarket cashiers, etc., or to computer and communication systems, e.g., multi-processor systems, data centers, etc. Scheduling in these complex systems concerns the assignment of a single server to execute each arriving job. Existing scheduling policies reveal an interesting tradeoff between 1) the optimality of some performance metric, e.g., jobs' (average) delay, and 2) cost efficiency, e.g., in terms of minimizing the amount of overhead. At one extreme, the policy of (non-)randomly selecting a server has no feedback cost (as communication from the servers to the dispatcher) but conceivably lends itself to very large delays, and even to instabilities when the selection process is not adequately balanced. At the other extreme, the *Join the Shortest Queue* (JSQ) policy, whereby the dispatcher sends each job to the server with the shortest queue, minimizes delay but has a very high feedback cost because all servers must report their queue lengths for every job arrival, and thus raises a valid concern regarding practical implementations.

In order to reduce the feedback cost, and yet to keep the delay 'small', JSQ has been generalized to SQ(d), whereby the dispatcher runs JSQ only for a *subset* of d randomly sampled servers from the uniform distribution (see Mitzenmacher [13] and Luczak and McDiarid [11]). Note that SQ(d) reduces to a simple uniform random selection when $d = 1$, and to JSQ when $d = N$, where N is the total number of servers. A fundamental qualitative result is that SQ(2) yields an exponential improvement over SQ(1) in terms of delay, yet with a conceivably small overhead cost. This result is known as the 'power-of-two' result [13] and was independently obtained by Vvedenskaya *et al.* [16].

Despite its apparent simplicity, SQ(d) is very difficult to analyze in terms of the delay metric, even for a classical input with Poisson arrivals and exponential job sizes. In fact, SQ(d) can only be exactly analyzed for $d = 1$, in which case the problem reduces to the M/M/1 queue. What makes the problem particularly difficult, when $d > 1$, is that the generator matrix of an underlying N-dimensional Markov process (representing, for instance, the number of jobs at each of the servers' queues) has an irregular structure. For this reason, solutions have so far been developed either in asymptotic regimes or in terms of bounds in particular cases.

An exact solution on the delay distribution was obtained in an asymptotic regime in the total number of servers, i.e., for $N \to \infty$ for Poisson arrivals and exponential service times [16], [13]; this solution was instrumental to showing the 'power-of-two' result. The extension to general service times was addressed in Bramson *et al.* [3], [4]. The corresponding case of the randomized longest-queue-first policy was addressed asymptotically in Alanyali and Dashouk [2] and Dieker and Suk [5]. The case of heterogenous servers, Poisson arrivals and general service time distributions was addressed under an Erlang loss server model by Mukhopadhyay *et al.* [14] and also under a light load regime with FCFS servers by Izagirre and Makowski [7].

As far as non-asymptotic solutions are concerned, upper and lower bounds on delay were obtained for the particular case when $d = N$, i.e., JSQ. The main idea is to transform the original Markov process with the inherent irregular structure into Markov chains with some regular structure (see Adan et al. [1], Lui et al. [12], or Zhao and Grassmann [17]). To get a lower bound, for instance, the transformation consists of redirecting some transitions between the states of the original Markov process in such a way that the new system is less loaded than the original one. Moreover, the newly formed generator matrix has a periodic structure such that its analysis becomes amenable to matrix-geometric techniques (Neuts [15]).

In this paper we extend such methods for computing upper and lower delay bounds to the general SQ(d) case. The extension is not straightforward, but on the contrary, because of a much more compounded transformation process needed to produce Markov processes with a regular structure. We thus provide the first non-asymptotic results for the SQ(d)

policy which can be applied in finite regimes with small to moderate number of servers. One drawback of the obtained bounds, however, is that they are obtained in implicit form, as they are based on matrix-geometric techniques, and are thus unable to provide qualitative insight alike the 'power-of-two' result. In terms of numerical accuracy, the lower bounds are remarkably tight; in turn, the upper bounds also become tight but only at an exponential cost in numerical complexity (by properly adjusting a model parameter).

The rest of the paper is organized as follows. We first describe the SQ(d) model together with the associated lower and upper bound models. In Section III we prove the corresponding stochastic ordering on the lower and upper models, relative to the base model. In Section IV we present a numerical analysis of the lower and upper bound models. Concrete numerical results are illustrated in Section V and brief conclusions are presented in Section VI.

II. The Model

We consider the general SQ(d) scheduling policy with N parallel servers. Jobs arrive at a central dispatcher according to a Poisson process with rate λN, and their service times are exponentially distributed with unit mean. With every arriving job, the dispatcher randomly polls d servers according to a uniform distribution without replacement, out of the N servers. The d selected servers report the number of jobs in their systems, and the newly arriving job joins the server with the smallest number of existing jobs; ties are resolved arbitrarily (see Figure 1). At every server, jobs are served according to the FIFO policy. We enforce the stability condition $\lambda < 1$.

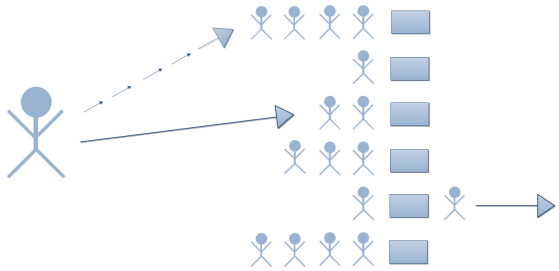


Fig. 1. The SQ(d) model with $N = 6$ servers and $d = 2$ choices; the arriving job joins server 3 (counted from top to bottom); the departing job from server 5 is arbitrary.

The Poisson/exponential arrivals' model enables the construction of a continuous-time Markov process to model the evolution of the SQ(d) policy. The set of states is

$$\mathcal{M} = \{\boldsymbol{m} : \boldsymbol{m} = (m_1, m_2, \ldots, m_N)\} \ , \qquad (1)$$

where m_1 denotes the largest number of jobs at the N servers, m_2 denotes the second largest number of jobs, and so on, such that m_N denotes the smallest number of jobs.

A. Transition Rates

Consider a generic state $\boldsymbol{m} \in \mathcal{M}$. We distinguish two cases, depending on the uniqueness of $\mathbf{m}$'s components. In the first case, all the servers have distinct numbers of jobs, such that the elements of $\mathbf{m}$ can be written as

$$m_1 > m_2 > \cdots > m_N \ .$$

The transition rates are in this case

$$\begin{aligned} \lambda(\boldsymbol{m}, \boldsymbol{m} + \boldsymbol{e_i}) &= \frac{\binom{i-1}{d-1}}{\binom{N}{d}} \lambda N, \ \forall d \leq i \leq N, \text{ and} \\ \mu(\boldsymbol{m}, \boldsymbol{m} - \boldsymbol{e_i}) &= \mu \ \forall 1 \leq i \leq N \ , \end{aligned}$$

where $\lambda(\boldsymbol{m}, \boldsymbol{m} + \boldsymbol{e_i})$ and $\mu(\boldsymbol{m}, \boldsymbol{m} - \boldsymbol{e_i})$ are the transition rates from state $\boldsymbol{m}$ to the states $\boldsymbol{m} + \boldsymbol{e_i}$ and $\boldsymbol{m} - \boldsymbol{e_i}$, respectively. Here, $\boldsymbol{e_i}$ is defined as the unit vector containing only zeros, except for the i^{th} element which is set to one.

It is instructive to explain the transition rate $\lambda(\boldsymbol{m}, \boldsymbol{m} + \boldsymbol{e_i})$ corresponding to an arriving job. According to the SQ(d) policy, there are $\binom{N}{d}$ distinct ways to poll d servers out of a total of N servers. Moreover, in order for the i^{th} server to be eventually selected, it must be polled by the dispatcher, and in addition $d-1$ other servers with greater number of jobs than server i must be polled as well; since there are at most $i-1$ such servers, we get the binomial factor $\binom{i-1}{d-1}$ from above. Note that, as expected, $\sum_{i=d}^{N} \binom{i-1}{d-1} = \binom{N}{d}$. In turn, the other transition rate $\mu(\boldsymbol{m}, \boldsymbol{m} - \boldsymbol{e_i})$ corresponding to a departing job follows immediately by the assumption of exponential service times.

The other slightly more complicated case is when at least two of the servers have an equal number of jobs. There exists thus $1 \leq i \leq N$ and $j > 1$ such that the elements of $\mathbf{m}$ can be written as

$$m_1 \geq \ldots \geq m_{i-1} > m_i = \ldots = m_{i+j} > m_{i+j+1} \geq \ldots \geq m_N \ .$$

Let us now make two important conventions. If a server k with $i \leq k \leq i + j$ is being polled, and its number of jobs is smaller than at the other $d - 1$ servers being polled, then we reorder the elements of $\mathbf{m}$ such that it appears as if server i had been selected. The other convention is that if a job departs from the server k, then we reorder the elements of $\mathbf{m}$ such that it appears as if the job had departed from server $i + j$. We point out that these two conventions do not change the system, but they are simply made for imposing a convenient ordering of the elements of $\mathcal{M}$.

According to the first convention, we have the following transitions for arrivals

$$\begin{aligned} \lambda(\boldsymbol{m}, \boldsymbol{m} + \boldsymbol{e_i}) &= \frac{\sum_{k=i}^{i+j} \binom{k-1}{d-1}}{\binom{N}{d}} \lambda N = \frac{\binom{i+j}{d} - \binom{i-1}{d}}{\binom{N}{d}} \lambda N \\ \lambda(\boldsymbol{m}, \boldsymbol{m} + \boldsymbol{e_k}) &= 0 \ \forall i + 1 \leq k \leq i + j \ . \end{aligned}$$

There are two interpretations for the numerator in the first transition. One is that any of the servers $i \leq k \leq i + j$ must be polled, and in addition $d-1$ out of the servers $1 \leq l \leq k-1$ must be polled as well. The other interpretation is that all d servers must be polled out of the first $i + j$ servers, and at least one must be polled out of the servers $i \leq k \leq i + j$. We remark that, unlike in the case with distinct number of jobs

across all the servers, the arrival rates do depend now on the number of servers with equal number of jobs.

Concerning the departure rates, we have according to the second convention from above

$$\begin{aligned} \mu(\boldsymbol{m}, \boldsymbol{m} - \boldsymbol{e_{i+j}}) &= (j+1)\mu \\ \mu(\boldsymbol{m}, \boldsymbol{m} - \boldsymbol{e_k}) &= 0 \ \forall i \leq k \leq i+j-1 \ . \end{aligned}$$

The previously defined arrival and departure transition rates form the generator matrix Q of the underlying Markov process describing the evolution of the SQ(d) policy. In order to analyze this policy, e.g., in terms of the jobs' steady-state average delay, one should compute the equilibrium probabilities $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots)$ associated to the states of the Markov process, by solving for

$$\boldsymbol{\pi} Q = \mathbf{0} \text{ and } \boldsymbol{\pi} \boldsymbol{e} = 1 \ , \tag{2}$$

where $\mathbf{0}$ denotes the all 0's vector and $\boldsymbol{e}$ denotes the all 1's vector.

Although the existence of a steady-state distribution π is guaranteed by the fact that all states of the underlying irreducible Markov process are positive recurrent, its explicit computation is hampered by the irregular structure of the generator matrix Q. By 'irregular' we mean that there is no apparent periodic or recursive generic representation for the infinite sized matrix Q which would allow for solving the system of equations from Eq. (2). To illustrate the irregularity of Q, we next show a finite chunk of the transition flow diagram of the SQ(d) model with $N = 3$ servers and $d = 2$ choices (see Figure 2).

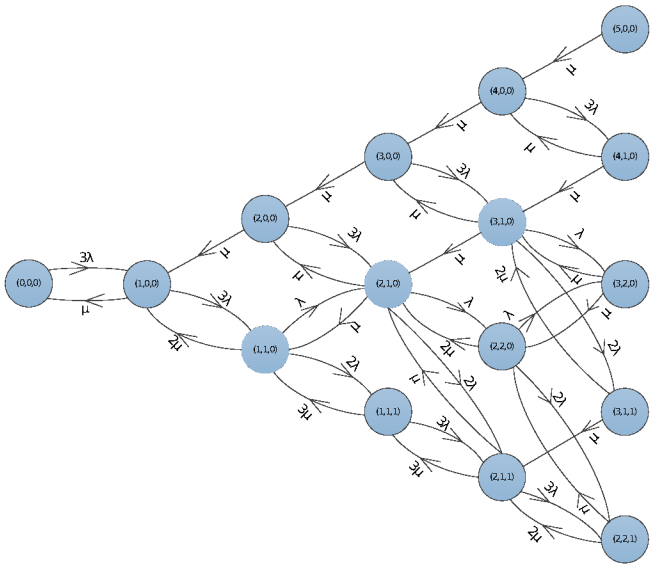


Fig. 2. A chunk of the transition flow diagram of the SQ(2) policy, with $N = 3$ servers

B. SQ(d) Lower and Upper Bound Models

In order to circumvent the irregularity structure of Q, for our purpose of computing the jobs' average delay, we extend the ideas from the JSQ analysis (see Adan et al. [1]), which is a particular case of SQ(d). Concretely, we are going to make two transformations of the original Markov process by suitably redirecting its transitions such that the new generator matrices have some regular structure, and whose analysis can be tackled with matrix analytic methods. The transformations are such that the average delays in the first and second transformed models are lower and upper bounds, respectively, for the average delay in the original model.

Both transformations have in common a threshold parameter T and the following condition

$$m_1 - m_N \leq T \ , \tag{3}$$

which must hold for the elements of $\mathbf{m}$ defined in Eq. (1) for both transformed Markov processes (one for getting lower bounds and the other for getting upper bounds). For convenience, we refer to the two transformed models as the lower and upper bound models, respectively. As expected, the parameter T adjusts the accuracy of the models; in particular, higher values yield tighter bounds, at the expense however of an increasing numerical complexity.

Next we show the constructions of these models by suitably redirecting some of the transitions of the original Markov process in order to enforce the condition from Eq. (3).

1) Lower Bound: The transitions are redirected according to the following two rules, one for arrivals, and the other for departures.

1) When an arrival causes the violation of Eq. (3), then the arrival is sent to (one of) the shortest queue(s) instead of the longest queue.
2) When a departure causes the violation of Eq. (3), then the departure occurs from (one of) the longest queue(s) instead of the shortest queue.

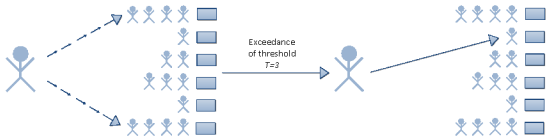


Fig. 3. Arrival redirection for the lower bound model; SQ(2), $N = 6$ servers, and threshold $T = 3$

Figure 3 illustrates a redirected arrival from either server 1 or 6 (counted in the figure from top to bottom), which would cause the condition from Eq. (3) to fail, to server 2 (which is one of the two with the smallest number of jobs).

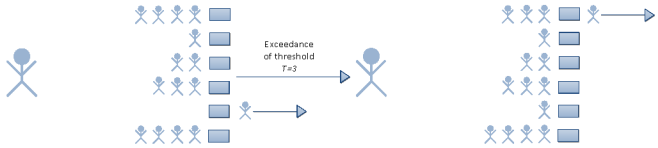


Fig. 4. Departure redirection for the lower bound model; SQ(2), $N = 6$ servers, and threshold $T = 3$

Figure 4 illustrates a redirected departure from server 5, which would cause the condition from Eq. (3) to fail, to server 1 (which is one of the two with the biggest number of jobs).

2) Upper bound: The transitions are redirected according to the following two rules:

1) When an arrival causes the violation of Eq. (3), then the arrival is accompanied by the addition of one extra job at each of the shortest queues.
2) When a departure causes the violation of Eq. (3), then the departure may not occur.

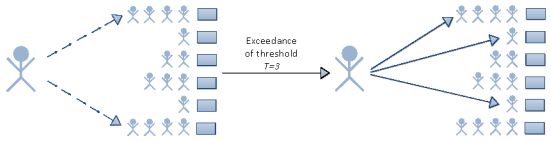


Fig. 5. Arrival(s) addition for the upper bound model; SQ(2), $N = 6$ servers, and threshold $T = 3$

In Figure 5, the selection of server 1 and 6 would cause Eq. (3) to fail; therefore, the arbitrary selection of server 1 is accompanied by the (artificial) addition of two more jobs at servers 2 and 5, which have the smallest number of jobs.

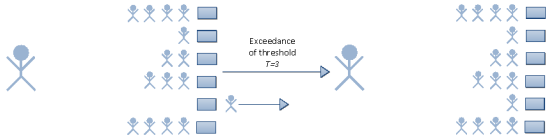


Fig. 6. Departure cancellation for the upper bound model; SQ(2), $N = 6$ servers, and threshold $T = 3$

Figure 6 illustrates that a departure from server 5, which would cause Eq. (3) to fail, is cancelled.

3) Achieving regularity: The key advantage of the previous model transformations is that they eliminate the irregular structure of the original generator matrix Q, and they lend themselves to new generator matrices with some regular structures.

To provide some intuition on this fact, let us refer to Figure 7 displaying (a chunk of) the transition flow diagram for the SQ(d) lower bound model with $N = 3$ servers, $d = 2$ choices, and threshold $T = 2$. The states with the same number of jobs are positioned on the same imaginary vertical line. An example of a *modified* transition is 3λ from $(2, 2, 0)$ to $(2, 2, 1)$. Note that, in the original model, there is a 2λ (rate) transition to $(2, 2, 1)$ and a λ transition to $(3, 2, 0)$; because the state $(3, 2, 0)$ violates the threshold condition, the λ transition is redirected to $(2, 2, 1)$, and thus the 3λ transition. Another example of a modified transition is 3μ from $(3, 1, 1)$ to $(2, 1, 1)$, which is formed from a single μ from $(3, 1, 1)$ to $(2, 1, 1)$ (in the original model) and 2μ by redirecting departures from the two servers with only one job to the server with 3 jobs.

What is important to remark is that, after some initial stage, the structure of the transition flow diagram repeats itself. Indeed, by observing the set of states with 5, 6, and 7 number of jobs in the system, we remark the same pattern of transition rates as in the set of states with 8, 9, and 10 number of jobs in the system; in fact, this patterns repeats indefinitely. A similar regularity pattern can be observed for the transformed upper bound model (see Figure 8).

The regularity of the constructed generator matrices, in both the lower and upper bound models, will enable the computation of lower and upper bounds on the jobs' average delay by relying on matrix analytical methods (see Section IV). Before doing that, however, we prove that the delays in the transformed models are indeed lower and upper bounds for the original one.

III. Stochastic Ordering

In this section we prove that the previous modified models do lend themselves to lower and upper bound models for the original SQ(d) model. In particular, we will show the underlying stochastic ordering with respect to first moments on delays (stronger stochastic ordering results can also be shown to hold). Due to similarity, we only prove the upper bound.

The proof extends the one from Adan et al. [1] from JSQ to SQ(d), and is based on the idea of using cost functions, which are set such that the shorter/longer a job has to wait for its service completion, the lower/higher the costs are. Moreover, in our constructed lower and upper bound models, transitions are redirected such that the expected costs are always decreased or increased, respectively.

We now define the cost functions for the original, lower bound, and upper SQ(d) models. For the original SQ(d) model, we define $v_n(\boldsymbol{m})$ as the expected n-period costs, at the embedded transition points in the underlying Markov process, when starting in a state $\boldsymbol{m} \in \mathcal{M}$ (see Eq. (1)). Similarly, we define $u_n(\boldsymbol{m})$ and $w_n(\boldsymbol{m})$ as the expected n-period costs starting in state $\boldsymbol{m}$ for the SQ(d) lower and upper bound model, respectively. We assume for all three models and for all states $\boldsymbol{m}$ that $u_0(\boldsymbol{m}) = v_0(\boldsymbol{m}) = w_0(\boldsymbol{m}) = 0$.

The cost in a given state $\boldsymbol{m}$ is

$$c(\boldsymbol{m}) = \sum_i m_i \ ,$$

i.e., the total number of jobs in the system, which suffices for our purpose of proving the stochastic ordering in terms of average delays (as an application of Little's Law).

The key to the proof is to show that for all (relevant) states $\boldsymbol{m}$ and for all periods n

$$u_n(\boldsymbol{m}) \leq v_n(\boldsymbol{m}) \leq w_n(\boldsymbol{m}) \ . \tag{4}$$

Let us next focus on the upper bound (as mentioned, the proof for the lower bound is similar). We will proceed by induction on n. The case $n = 0$ holds by definition. Assuming that Eq. (4) holds for some $n \geq 0$ we will prove that it also holds for $n + 1$.

First we need a simplifying ordering result on *precedence pairs* of states from $\mathcal{M}$, ordered in a suitable manner. Using the notation $\boldsymbol{m} = (m_1, m_2, \ldots, m_N)$ and $\boldsymbol{m'} =$

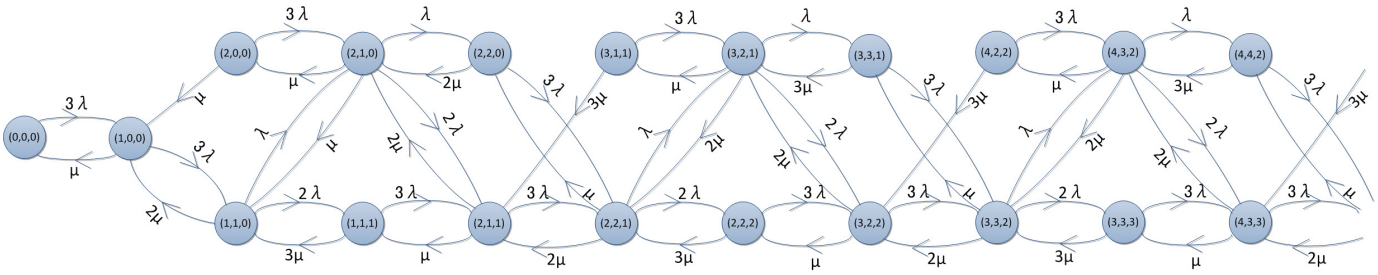


Fig. 7. Lower bound model; achieving regularity in the transition flow diagram (SQ(2), $N = 3$, and $T = 2$)

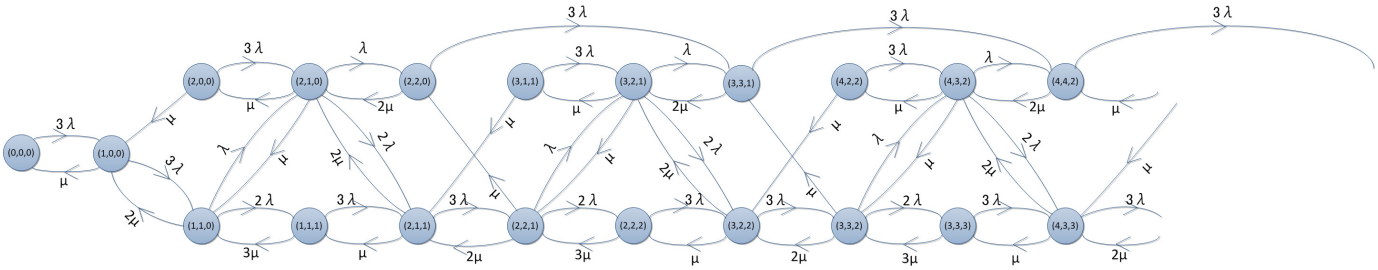


Fig. 8. Upper bound model; achieving regularity in the transition flow diagram (SQ(2), $N = 3$, and $T = 2$)

$(m'_1, m'_2, \ldots, m'_N)$ for some states from $\mathcal{M}$, we define the set of precedence pairs

$$P = \left\{ (\boldsymbol{m}, \boldsymbol{m'}) : \sum_{i=1}^{j} m_i \leq \sum_{i=1}^{j} m'_i \ \forall j = 1, \ldots, N \right\} . \quad (5)$$

Intuitively, the previous inequalities can be interpreted as being 'more preferable' to have less jobs in the longest j queues in the system. On one hand, when there are fewer jobs in the system, the costs are expected to be lower (by the definition of the cost function). On the other hand, in a more balanced system, the efficiency of the servers is improved and hence the costs are expected to decrease.

Let us next define $P_{\boldsymbol{m}}$ as the subset of precedence pairs $(\boldsymbol{m},\boldsymbol{m'})$ from P for which $\boldsymbol{m'}$ is equal to either $\boldsymbol{m}+\boldsymbol{e_N}, \boldsymbol{m}+\boldsymbol{e_1}-\boldsymbol{e_2}, \boldsymbol{m}+\boldsymbol{e_2}-\boldsymbol{e_3}, \ldots$, or $\boldsymbol{m}+\boldsymbol{e_{N-1}}-\boldsymbol{e_N}$.

For some precedence pair $(\boldsymbol{m},\boldsymbol{m'})$ let us observe that by defining

$$d_i = m'_i - m_i \ \forall 1 \leq i \leq N ,$$

and the associated partial sums

$$s_j = \sum_{i=1}^{j} d_i \ \forall 1 \leq j \leq N ,$$

then one can write

$$\boldsymbol{m'} = \boldsymbol{m} + s_N \boldsymbol{e_N} + s_{N-1}(\boldsymbol{e_{N-1}} - \boldsymbol{e_N}) + \cdots + s_1(\boldsymbol{e_1} - \boldsymbol{e_2}) . \quad (6)$$

In other words, any precedence pair $(\boldsymbol{m},\boldsymbol{m'})$ from P can be recursively obtained using precedence pairs from $P_{\boldsymbol{m_l}}$, for some states $\boldsymbol{m_l}$, $l = 1, \ldots, L$, with $\boldsymbol{m_1} = \boldsymbol{m}$ and $\boldsymbol{m_L} = \boldsymbol{m'}$.

Next we state without proof that for any precedence pair $(\boldsymbol{m},\boldsymbol{m'})$ in $P_{\boldsymbol{m}}$ it holds that

$$v_n(\boldsymbol{m}) \leq v_n(\boldsymbol{m'}), \ \forall n \geq 0 . \quad (7)$$

The proof is very tedious and similar to the one from [1] (constructed for the JSQ model), for which reason we omit it here. Note that the construction from Eq. (6) implies that Eq. (7) extends to the whole set of precedence pairs P.

Provided the modified chain has been constructed by redirecting transitions to less attractive states (i.e., a transition to $\boldsymbol{m'}$ is redirected to $\tilde{\boldsymbol{m}}'$ with $v_n(\boldsymbol{m'}) \leq v_n(\tilde{\boldsymbol{m}}')$), we have

$$\begin{aligned} v_{n+1}(\boldsymbol{m}) &= c(\boldsymbol{m}) + \sum_{\boldsymbol{m'}} p(\boldsymbol{m}, \boldsymbol{m'}) v_n(\boldsymbol{m'}) \\ &\leq c(\boldsymbol{m}) + \sum_{\tilde{\boldsymbol{m}}'} p(\boldsymbol{m}, \tilde{\boldsymbol{m}}') v_n(\tilde{\boldsymbol{m}}') \\ &\leq c(\boldsymbol{m}) + \sum_{\tilde{\boldsymbol{m}}'} p(\boldsymbol{m}, \tilde{\boldsymbol{m}}') w_n(\tilde{\boldsymbol{m}}') \\ &= w_{n+1}(\boldsymbol{m}) , \end{aligned}$$

where the second inequality follows from the induction hypothesis, and thus completing the induction proof.

IV. Numerical Analysis

In this section we numerically analyze the SQ(d) upper and lower bound models, which provide stochastic bounds for the original SQ(d) model. We first present a numerical method to compute the mean waiting time of jobs for the SQ(d) upper bound model. While this method applies for the SQ(d) lower bound model as well, we will later present an improved method for the latter (see Section IV-B).

A. Upper and Lower Bounds

As we have seen in Section II, the transition flow diagrams of the SQ(d) lower and upper bound models are, in contrast to the transition flow diagram of SQ(d), well structured. The key advantage of these transformed models is that we can partition the newly constructed state spaces (for the lower/upper bounds systems) into blocks of states with a periodic structure between adjacent blocks. Moreover, each block has a finite number of states which can be further ordered according to the total number of jobs in the system; ties are broken according to a lexicographical ordering. Concretely, the first block of states is defined as

$$B_{\leq(N-1)T} = \{\boldsymbol{m} \in S \mid \#\boldsymbol{m} \leq (N-1)T\} \ , \qquad (8)$$

and corresponds to the boundary states. Here $\#\boldsymbol{m}$ is defined as the number of jobs in state $\boldsymbol{m}$. Observe that all states for which $m_N = 0$ are included in the boundary states. All the states with the same number of jobs are grouped together. Moreover, all boundary states are those states $\boldsymbol{m'}$ for which there is a state $\boldsymbol{m}$ with $\#\boldsymbol{m} = \#\boldsymbol{m'}$ and $m_N = 0$. The state with the most number of jobs in the system and with $m_N = 0$ is the state (T, T, ..., T, 0). As there are N servers, the total number of jobs in this state is $(N-1)T$. Therefore, Eq. (8) corresponds to the set of boundary states.

For the rest of the state space we define the blocks

$$B_q = \{\boldsymbol{m} \in S | (N-1)T + qN < \#\boldsymbol{m} \leq (N-1)T + (q+1)N\}, \ \forall q = 0, 1, 2, \ldots$$

Note that we expect a regular pattern as we have seen in the transition flow diagrams from Figures 7 and 8. In particular, except for boundary states,

$$p_{\boldsymbol{m},\boldsymbol{m'}} = p_{\boldsymbol{m}+\boldsymbol{1},\boldsymbol{m'}+\boldsymbol{1}} \ . \qquad (9)$$

(see the 'Useful Properties' in the Appendix.) The difference between these two transition probabilities stems from the states which are involved. Furthermore, the difference between the corresponding states is the number of jobs at each server, which is one. The difference in the total number of jobs in the system is therefore N. We thus expect that every state in set B_q will correspond to exactly one state in set B_{q+1} for all nonnegative integers q. Moreover, every state in set B_q will correspond to exactly one state in set B_{q+l} for all nonnegative integers q and all nonnegative integers l.

Having partitioned the state space into blocks of states, i.e.,

$$S = B_{\leq(N-1)T} \cup (\cup_{q=0}^{\infty} B_q) \ ,$$

we are now ready to construct the newly generator matrices Q of the upper and lower bound models. They have the following structured form

$$Q = \begin{pmatrix} R_{00} & R_{01} & 0 & 0 & 0 & \ldots \\ R_{10} & A_1 & A_0 & 0 & 0 & \ldots \\ 0 & A_2 & A_1 & A_0 & 0 & \ldots \\ 0 & 0 & A_2 & A_1 & A_0 & \ldots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix} \ .$$

Here, R_{00}, R_{01} and R_{10} correspond to the matrices created by transition rates within the boundary blocks, transitions from a non-boundary block to a boundary block and from a boundary block to a non-boundary block, respectively. The non-boundary blocks, i.e., A_0, A_1 and A_2, are of order $m \times m$, where m is the number of states in such a block, i.e.,

$$m = \binom{N+T-1}{T} \ .$$

Given the property from Eq. (9), all submatrices on the main diagonal are, except for R_{00}, identical. We call this submatrix A_1. Also, all submatrices on the subdiagonal on the left and all submatrices on the right of the main diagonal are, except for R_{10} and R_{01}, identical, respectively. We call these matrices A_2 and A_0, respectively. Because of this structure of the generator matrix Q, the stationary equations are given by

$$(\boldsymbol{\pi}_{\leq(N-1)T}, \boldsymbol{\pi}_0, \boldsymbol{\pi}_q)Q = \mathbf{0} \ ,$$

where $\boldsymbol{\pi}_{\leq(N-1)T}$ is the limiting probability of the boundary block and $\boldsymbol{\pi}_q$ is the limiting probability of block B_q for all $q \in \{0, 1, 2, \ldots\}$. On blocks, we can write the balance equations for the equilibrium probabilities as

$$\mathbf{0} = \boldsymbol{\pi}_{\leq(N-1)T} R_{00} + \boldsymbol{\pi}_0 R_{10} \qquad (10)$$

$$\mathbf{0} = \boldsymbol{\pi}_{\leq(N-1)T} R_{01} + \boldsymbol{\pi}_0 A_1 + \boldsymbol{\pi}_1 A_2 \qquad (11)$$

$$\mathbf{0} = \boldsymbol{\pi}_{q-1} A_0 + \boldsymbol{\pi}_q A_1 + \boldsymbol{\pi}_{q+1} A_2, \quad q = 1, 2, \ldots \qquad (12)$$

Eqs. (10) and (11) are called the boundary equations, and Eqs. (12) are called the queue equations.

For the analysis of the SQ(d) upper and lower bound models, we construct a matrix R whose elements R_{ij} are the expected number of visits to state j in block B_1, starting from state i in block B_0. This matrix R is called the *rate matrix* and is characterized by

$$\begin{aligned} \mathbf{0} &= \sum_{k=0}^{\infty} R^k A_k \\ &= A_0 + RA_1 + R^2 A_2 \end{aligned}$$

Note that R is an $m \times m$ matrix as the number of states in both B_0 and B_1 is $\binom{N+T-1}{T}$. In order to use matrix-geometric techniques, we observe that the generator matrix Q is irreducible, since the matrices B_0 and A_1 are non-singular (their determinant is not zero). Also, assuming a stability condition, all states are positive recurrent and, consequently, the generator matrix Q is positive recurrent. Therefore, we can use Theorem 1.7.1 from Neuts [15], which states that the solutions of the stationary probabilities of the SQ(d) lower and upper bound model can be obtained by solving the balance equations

$$(\boldsymbol{\pi}_{\leq(N-1)T}, \boldsymbol{\pi}_0, \boldsymbol{\pi}_1) \begin{pmatrix} R_{00} & R_{01} & 0 \\ R_{10} & A_1 & A_0 \\ 0 & A_2 & A_1 + RA_2 \end{pmatrix} = \mathbf{0} \qquad (13)$$

with the normalization condition

$$\boldsymbol{\pi}_{\leq(N-1)T} \boldsymbol{e} + (\boldsymbol{\pi}_0 + \boldsymbol{\pi}_1)(I - R)^{-1} \boldsymbol{e} = 1 \ ,$$

where $\boldsymbol{e}$ is the all one vector of proper dimensions.

For the lower bound model, the stability condition is $\lambda < 1$. However, for the SQ(d) upper bound model, this stability condition is no longer sufficient as the service capacity is reduced by the redirected transitions to less preferable states. Therefore, the balance equations from (13) only have a solution if and only if (see again Theorem 1.7.1 of Neuts [15])

$$\pi A_0 \boldsymbol{e} < \pi A_2 \boldsymbol{e} \ ,$$

where π is given by $\pi A = \mathbf{0}, \pi \boldsymbol{e} = 1$, and where $A = A_0 + A_1 + A_2$.

What remains to show is the construction of the rate matrix R, for which we use the technique described in Latouche and Ramaswami [9], [10]. Therein it is shown how to derive a matrix G, whose elements G_{ij} represent the probability that starting from a state i in block B_1 the chain eventually visits block B_0 and does so by visiting state j. As for the rate matrix R, the matrix G can be characterized by the following equation

$$\mathbf{0} = A_2 + A_1 G + A_0 G^2 \ .$$

The matrix G for a generator matrix Q is then explicitly given by

$$G = -\sum_{k=1}^{\infty} (\prod_{i=1}^{k} B_{1,i}) B_{2,k} \ ,$$

where

$$\begin{aligned} B_{1,1} &= (-A_1)^{-1} A_0 \\ B_{2,1} &= (-A_1)^{-1} A_2 \\ B_{1,i} &= (I - B_{1,i-1}B_{2,i-1} - B_{2,i-1}B_{1,i-1})^{-1} B_{1,i-1}^2 \\ B_{2,i} &= (I - B_{1,i-1}B_{2,i-1} - B_{2,i-1}B_{1,i-1})^{-1} B_{2,i-1}^2 \ . \end{aligned}$$

As a side remark, Latouche and Ramaswami [10] claim that the algorithm to compute G needs only few iterations k. We confirm this to hold for our system configurations, for which the number of iterations is within $k = 6$.

Finally, the rate matrix R can be computed from the matrix G by (see Latouche and Ramaswami [9])

$$R = -A_0 (A_1 + A_0 G)^{-1} \ .$$

Having the numerical algorithm to compute the rate matrix R, we are able to obtain the steady-state probabilities by solving the balance equations from (13) with the normalization condition. These lend themselves to stochastic lower and upper bound on the mean waiting time for the SQ(d) model. Concretely, for each state we know how many waiting jobs there are at each server, i.e., server i has $\max\{(m_i - 1), 0\}$ waiting jobs, and we can multiply this number by the equilibrium probability of the corresponding state. By doing so for all states, we can compute the jobs' average delay in a numerically tractable manner.

The above results are summarized in the following theorem.

Theorem 1. *The solutions of the stationary probabilities of the SQ(d) lower and upper bound model have a modified vector-geometric form. Specifically,*

$$\boldsymbol{\pi}_{q+1} = R \boldsymbol{\pi}_q, \quad q = 1, 2, \ldots$$

and $(\boldsymbol{\pi}_{\leq (N-1)T}, \boldsymbol{\pi}_0, \boldsymbol{\pi}_1)$ *can be obtained by solving the balance equations*

$$\begin{aligned} &(\boldsymbol{\pi}_{\leq (N-1)T}, \boldsymbol{\pi}_0, \boldsymbol{\pi}_1) \begin{pmatrix} R_{00} & R_{01} & 0 \\ R_{10} & A_1 & A_0 \\ 0 & A_2 & A_1 + RA_2 \end{pmatrix} \\ &\quad = (\boldsymbol{\pi}_{\leq (N-1)T}, \boldsymbol{\pi}_0, \boldsymbol{\pi}_1) \ . \end{aligned}$$

B. Improved Lower Bound

In the previous subsection we have obtained a numerically tractable method to compute the steady-state probabilities for the SQ(d) lower and upper bound models. Here we simplify this method dramatically, in terms of the numerical complexity, by demonstrating an important relation between steady-state probabilities, in the case of the lower bound model. Concretely, we will show that for non-boundary states $\boldsymbol{\pi}_{q+1} = \rho^N \boldsymbol{\pi}_q$, for all $q = 1, 2, \ldots$. Next we present this fundamental result for some arbitrary arrival process $A(t)$, and after we give a more explicit solution when $A(t)$ is a Poisson process.

Theorem 2. *The solutions of the stationary probabilities of the SQ(d) lower bound model have a modified vector-geometric form. Specifically,*

$$\boldsymbol{\pi}_{q+1} = \sigma^N \boldsymbol{\pi}_q, \quad q = 1, 2, \ldots$$

and $(\boldsymbol{\pi}_{\leq (N-1)T}, \boldsymbol{\pi}_0, \boldsymbol{\pi}_1)$ *can be obtained by solving the balance equations*

$$\begin{aligned} &(\boldsymbol{\pi}_{\leq (N-1)T}, \boldsymbol{\pi}_0, \boldsymbol{\pi}_1) \begin{pmatrix} R_{00} & R_{01} & 0 \\ R_{10} & A_1 & A_0 \\ 0 & A_2 & A_1 + \sigma^N A_2 \end{pmatrix} \\ &\quad = (\boldsymbol{\pi}_{\leq (N-1)T}, \boldsymbol{\pi}_0, \boldsymbol{\pi}_1) \ . \end{aligned} \tag{14}$$

Here σ is the unique solution for x, inside the unit circle, of the equation

$$x = \sum_{k \geq 0} x^k \beta_k \ , \tag{15}$$

and where

$$\beta_k = \int_0^{\infty} \frac{(\mu t)^k}{k!} e^{-\mu t} dA(t) \ .$$

For the proof see Godtschalk [6].

As our arrivals are assumed to be Poisson, we can give a stronger result for the SQ(d) lower bound model, in the sense that we can compute the solution σ for x.

Theorem 3. *In the case of Poisson arrivals, the solutions of the stationary probabilities of the SQ(d) lower bound model have the modified vector-geometric form*

$$\boldsymbol{\pi}_{q+1} = \rho^N \boldsymbol{\pi}_q \ , \quad q = 1, 2, \ldots$$

where ρ is the traffic intensity and the rest is as in Theorem 2.

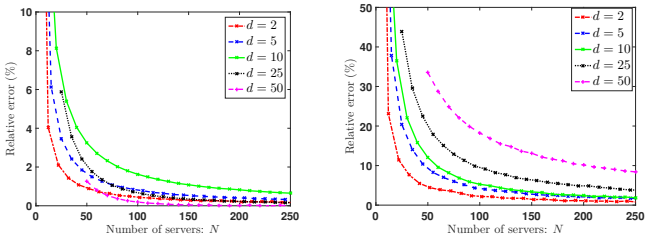
The proof follows using calculus methods and is deferred to the Appendix.

V. Numerical results

We first numerically motivate the need for addressing the problem of randomized load balancing in finite regimes. Consider the exact, but asymptotic result on the average delay in a SQ(d) system ([13]):

$$E[\text{Delay}] = \sum_{i=1}^{\infty} \lambda^{\frac{d^i - d}{d-1}} \ . \tag{16}$$

Note in particular that the expression is invariant to the number of servers n.

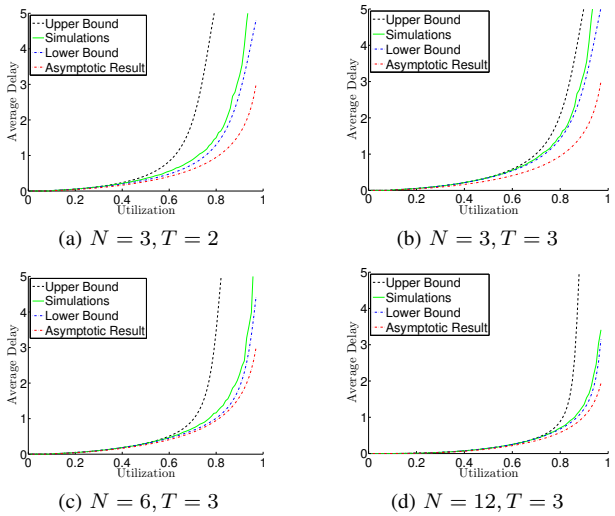


(a) $\rho = 0.75$ (b) $\rho = 0.95$

Fig. 9. Relative error of the predicted asymptotic delays vs. simulations

The accuracy of this approximation is illustrated in [13] for several values of n. Figure 9 herein provides further numerical results on the relative error (in %) of the asymptotic results from Eq. (16) relative to simulation results (obtained by simulating 10^8 jobs, out of which the first 10^7 were discarded). We consider two utilization levels $\rho = 0.75, 0.95$, several number of choices $d = 2, 5, 10, 25, 50$, and a broad range of number of servers n. The results clearly indicate that the approximation can be misleading in a regime with a small number of servers, and especially at very high utilizations. Another interesting observation is that the relative error is not necessarily monotonous in d, as shown in the moderately-high utilization case.

Next we illustrate the accuracy of our lower and upper bounds in the SQ(d) model. In Figure 10.(a-d) we show the average delay as a function of utilization for SQ(2). The first observation is that there is a tradeoff between the accuracy of the upper bounds and the computational complexity. Indeed, (a) and (b) indicate that the upper bounds are quite loose by letting $T = 2$, and are getting significantly tighter by letting $T = 3$. However, the numerical complexity increases significantly with T because the sizes of the (non-)boundary blocks in the generator matrix Q are exponential in T. As a related remark, different values of T change the stability condition for the SQ(d) upper bound (recall the last two rules for redirecting transitions from the previous section). The second observation is that the lower bounds are accurate over all three values of N, i.e., 3, 6, and 12. Finally, as also partially illustrated in Figure 9, the asymptotic results significantly underestimate the 'true' results for small values of N, and especially at high utilizations.



(a) $N = 3, T = 2$ (b) $N = 3, T = 3$

(c) $N = 6, T = 3$ (d) $N = 12, T = 3$

Fig. 10. Average delay as a function of utilization for SQ(2); various number of servers $N = 3, 6, 12$ and threshold parameter $T = 2, 3$

VI. Conclusions and Future Work

In this paper we have considered the SQ(d) scheduling policy whose analysis has so far been restricted to asymptotic regimes in the number of servers. Our central idea was to artificially construct two scheduling models which provide stochastic upper and lower bounds for the average delay in the original SQ(d) model. The merit of the obtained bounds is that they hold in non-asymptotic/finite regimes, and thus complement the existing exact but asymptotic results. Numerical evaluations revealed that there is an interesting tradeoff between the accuracy of the obtained upper bounds and the dimension of the computational complexity. Moreover, the lower bounds are remarkably tight, whereas existing asymptotic approximations may be misleading in finite regimes, especially at very high utilizations.

A major constraint of our results, alike existing asymptotic ones, is the Poisson assumption on the arrivals which may conceivably provide poor estimates in the context of fitting real traces. For this reason, a potential and significant advantage of the matrix-geometric methodology employed in this paper is that it can be extended to the broad class of Markov Arrival Processes (MAP) and Phase-Type (PH) service distributions (see, e.g., Lakatos *et al.* [8]). Another valuable extension is the analytical understanding of the tradeoff between computational complexity and the accuracy of the bounds, in particular the upper ones.

Acknowledgements

The authors thank Sem Borst for his support, and also the anonymous reviewers for their well-thought suggestions. This work was partially funded by Telekom Innovation Laboratories and the DFG grant Ci 195/1-1.

REFERENCES

[1] I. J. B. F. Adan, G. J. van Houtum, and J. van der Wal. Upper and lower bounds for the waiting time in the symmetric shortest queue system. *Annals of Operations Research*, 48(2):197–217, Apr. 1994.

[2] M. Alanyali and M. Dashouk. Occupancy distributions of homogeneous queueing systems under opportunistic scheduling. *IEEE Transactions on Information Theory*, 57(1):256–266, Jan 2011.

[3] M. Bramson, Y. Lu, and B. Prabhakar. Randomized load balancing with general service time distributions. In *ACM Sigmetrics*, pages 275–286, 2010.

[4] M. Bramson, Y. Lu, and B. Prabhakar. Asymptotic independence of queues under randomized load balancing. *Queueing Systems*, 71(3):247–292, 2012.

[5] A. B. Dieker and T. Suk. Randomized longest-queue-first scheduling for large-scale buffered systems. *ArXiv e-prints*, June 2013.

[6] A. S. Godtschalk. Stochastic bounds for randomized load balancing. *MSc Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands*, 2012.

[7] A. Izagirre and A. M. Makowski. Light traffic performance under the power of two load balancing strategy: The case of server heterogeneity. *Performance Evaluation Review*, 42(2):18–20, Sept. 2014.

[8] L. Lakatos, L. Szeidl, and M. Telek. *Introduction to Queueing Systems with Telecommunication Applications*. Springer, 2012.

[9] G. Latouche and V. Ramaswami. A general class of Markov processes with explicit matrix-geometric solutions. *Springer-Verlag, OR Spektrum*, 8(4):209–218, Aug. 1986.

[10] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for quasi-birth-death processes. *Journal of Applied Probability*, 30(3):650–674, Sept. 1993.

[11] M. J. Luczak and C. McDiarmid. On the maximum queue length in the supermarket model. *Annals of Probability*, 34(2):493–527, May 2006.

[12] J. Lui, R. R. Muntz, and D. Towsley. Bounding the mean response time of the minimum expected delay routing policy: An algorithmic approach. *IEEE Transactions on Computers*, 44(12):1371–1382, Dec. 1995.

[13] M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1094–1104, Oct. 2001.

[14] A. Mukhopadhyay, R. R. Mazumdar, and F. Guillemin. The power of randomized routing in heterogeneous loss systems. In *International Teletraffic Congress (ITC 27)*, pages 125–133, Sept 2015.

[15] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach*. John Hopkins University Press, 1981.

[16] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problems of Information Transmission*, 32(1):15–29, 1996.

[17] Y. Zhao and W. K. Grassmann. Queueing analysis of a jockeying model. *JSTOR: Operations Research*, 43(3):520–530, May 1995.

APPENDIX

USEFUL PROPERTIES

Here we give two useful properties amongst the transition probabilities $p_{\boldsymbol{m},\boldsymbol{m'}}$, over the state space

$$S = \{\boldsymbol{m} = (m_1, \ldots, m_N) \mid m_1 \geq m_2 \ldots \geq m_N \text{ and } |m_1 - m_N| \leq T\} \quad .$$

Define t_l as the time just before the lth arrival and consider $\{\vec{X}_l = (X_1(t_l), X_2(t_l), \ldots, X_N(t_l)); l = 1, 2, \ldots\}$, which is, alike $\vec{X}_t = (X_1(t), X_2(t), \ldots, X_N(t))$, a Markovian process. Now we are able to express the transition probabilities by conditioning on the interarrival time U_l, i.e.,

$$p_{\boldsymbol{m},\boldsymbol{m'}} = \int_0^\infty P(\vec{X}_{l+1} = \boldsymbol{m'} | U_l = t, \vec{X}_l = \boldsymbol{m}) dA(t) \ .$$

For a state $\boldsymbol{m}$, recall that $\#\boldsymbol{m}$ denotes the total number of corresponding jobs, including both the jobs in service and the waiting jobs. We see that $p_{\boldsymbol{m},\boldsymbol{m'}} = 0$ if $\#\boldsymbol{m'} > \#\boldsymbol{m} + 1$, because we only consider single arrivals. The next two lemmas are useful for the proving the main results in the paper.

Lemma 1. *Let $\boldsymbol{m},\boldsymbol{m'} \in S$. If $\#\boldsymbol{m'} = \#\boldsymbol{m} + 1$ and $m_N > 0$ or if $\#\boldsymbol{m'} < \#\boldsymbol{m} + 1$ and $m_1 > T$, then*

$$p_{\boldsymbol{m},\boldsymbol{m'}} = p_{\boldsymbol{m}+\boldsymbol{1},\boldsymbol{m'}+\boldsymbol{1}} \ , \tag{17}$$

where $\mathbf{1} = (1, 1, \ldots, 1)$.

Proof. Define $\vec{X}'_l$ to be the state immediately after the arrival of the lth job. The proof considers two cases, i.e., $\#\boldsymbol{m'} = \#\boldsymbol{m} + 1, m_N > 0$ and $\#\boldsymbol{m'} < \#\boldsymbol{m} + 1, m_1 > T$.

For the first case we can write

$$\begin{aligned}
p_{\boldsymbol{m},\boldsymbol{m'}} &= \int_0^\infty P(\vec{X}_{l+1} = \boldsymbol{m'} | U_l = t, \vec{X}_l = \boldsymbol{m}) dA(t) \\
&= \int_0^\infty P(\vec{X}'_l = \boldsymbol{m'} | \vec{X}_l = \boldsymbol{m}) \\
&\quad P(\vec{X}_{l+1} = \boldsymbol{m'} | U_l = t, \vec{X}'_l = \boldsymbol{m'}) dA(t) \\
&= \int_0^\infty P(\vec{X}'_l = \boldsymbol{m'} | \vec{X}_l = \boldsymbol{m}) \\
&\quad P(\text{no job served} | U_l = t, \text{all servers busy at } t_l) dA(t) \\
&= \int_0^\infty P(\vec{X}'_l = \boldsymbol{m'} + \boldsymbol{1} | \vec{X}_l = \boldsymbol{m} + \boldsymbol{1}) \mathrm{e}^{-\mu t} dA(t) \\
&= p_{\boldsymbol{m}+\boldsymbol{1},\boldsymbol{m'}+\boldsymbol{1}}
\end{aligned}$$

Note that $\vec{X}'_l = \vec{X}_l + 1$ and also $\vec{X}_{l+1} = \vec{X}_l + 1$ as in this case there is only one arrival and no departure.

For the second case, we define the variable k to be $(\#\boldsymbol{m} + 1) - \#\boldsymbol{m'}$. Also define $\rightarrow \boldsymbol{m_1} \rightarrow \boldsymbol{m_2} \ldots \rightarrow \boldsymbol{m_k}$ as the event that the system is in state $\boldsymbol{m_1}$ after the first job is served, in state $\boldsymbol{m_2}$ after the second job is served, ..., and in state $\boldsymbol{m_k}$ after the kth job is served. We can write

$$\begin{aligned}
p_{\boldsymbol{m},\boldsymbol{m'}} &= \int_0^\infty \sum_{\boldsymbol{m_1},\ldots,\boldsymbol{m_k}:E} P(\rightarrow \boldsymbol{m_1} \ldots \rightarrow \boldsymbol{m_k} = \vec{X}_{l+1} \\
&\quad = \boldsymbol{m'} | U_l = t, \vec{X}_l = \boldsymbol{m}) dA(t) \\
&= \int_0^\infty \sum_{\boldsymbol{m_1},\ldots,\boldsymbol{m_k}:E} P(\rightarrow \boldsymbol{m_1} + \boldsymbol{1} \ldots \rightarrow \boldsymbol{m_k} + \boldsymbol{1} \\
&\quad = \vec{X}_{l+1} = \boldsymbol{m'} + \boldsymbol{1} | U_l = t, \vec{X}_l = \boldsymbol{m} + \boldsymbol{1}) dA(t) \\
&= p_{\boldsymbol{m}+\boldsymbol{1},\boldsymbol{m'}+\boldsymbol{1}} \ ,
\end{aligned}$$

where E denotes the event $\{\#\boldsymbol{m_1} = \#\boldsymbol{m}, \#\boldsymbol{m_2} = \#\boldsymbol{m} - 1, \ldots, \#\boldsymbol{m_k} = \#\boldsymbol{m} - k = \#\boldsymbol{m'}\}$. □

Lemma 2. *Let $\boldsymbol{m},\boldsymbol{m'} \in S$. If $\#\boldsymbol{m'} = \#\boldsymbol{m} + 1$ and $m_N > 0$, then*

$$\sum_{\substack{\boldsymbol{m'}: \\ \#\boldsymbol{m'} = \#\boldsymbol{m}+1}} p_{\boldsymbol{m},\boldsymbol{m'}} = \beta_0 \ ,$$

where

$$\beta_0 = \int_0^\infty e^{-\mu t} dA(t) \ .$$

If $\#\boldsymbol{m'} < \#\boldsymbol{m}+1$ with $m_1 > T$, define k to be $(\#\boldsymbol{m}+1) - \#\boldsymbol{m'}$. Then

$$\sum_{\substack{\boldsymbol{m'}: \\ \#\boldsymbol{m'}=\#\boldsymbol{m}+1-k}} p_{\boldsymbol{m},\boldsymbol{m'}} = \beta_k \ , \tag{18}$$

where

$$\beta_k = \int_0^\infty \frac{(\mu t)^k}{k!} e^{-\mu t} dA(t) \ . \tag{19}$$

Proof. As in the previous proof, we start with the case $\#\boldsymbol{m'} = \#\boldsymbol{m}+1$. From Eq. (17) and

$$\sum_{\boldsymbol{m}} P(\vec{X}_l' = \boldsymbol{m'} | \vec{X}_l = \boldsymbol{m}) = 1$$

it follows that

$$\begin{aligned}
\sum_{\substack{\boldsymbol{m'}: \\ \#\boldsymbol{m'}=\#\boldsymbol{m}+1}} p_{\boldsymbol{m},\boldsymbol{m'}} &= \sum_{\substack{\boldsymbol{m'}: \\ \#\boldsymbol{m'}=\#\boldsymbol{m}+1}} \int_0^\infty P(\vec{X}_l' = \boldsymbol{m} | \vec{X}_l = \boldsymbol{m}) \mathrm{e}^{-\mu t} dA(t) \\
&= \int_0^\infty \sum_{\substack{\boldsymbol{m'}: \\ \#\boldsymbol{m'}=\#\boldsymbol{m}+1}} P(\vec{X}_l' = \boldsymbol{m} | \vec{X}_l = \boldsymbol{m}) \mathrm{e}^{-\mu t} dA(t) \\
&= \int_0^\infty \mathrm{e}^{-\mu t} dA(t) = \beta_0
\end{aligned}$$

For the second case $\#\boldsymbol{m'} < \#\boldsymbol{m}+1, m_1 > T$, we have from Eq. (17)

$$\begin{aligned}
\sum_{\substack{\boldsymbol{m'}: \\ \#\boldsymbol{m'}=\#\boldsymbol{m}+1-k}} p_{\boldsymbol{m},\boldsymbol{m'}} &= \sum_{\boldsymbol{m_1},\ldots,\boldsymbol{m_k}:E} \int_0^\infty P(\rightarrow \boldsymbol{m_1} \ldots \rightarrow \boldsymbol{m_k} | U_l = t, \vec{X}_l = \boldsymbol{m}) dA(t) \\
&= \int_0^\infty \sum_{\boldsymbol{m_1},\ldots,\boldsymbol{m_k}:E} P(\rightarrow \boldsymbol{m_1} \ldots \rightarrow \boldsymbol{m_k} | U_l = t, \vec{X}_l = \boldsymbol{m}) dA(t) \\
&= \int_0^\infty P(k \text{ jobs served } | U_l = t, \text{ all servers busy at } t_l) dA(t) \\
&= \int_0^\infty \frac{(\mu t)^k}{k!} \mathrm{e}^{-\mu t} dA(t) \\
&= \beta_k \ .
\end{aligned}$$

□

PROOF of Theorem 3

The proof is similar to the one of Theorem 2. What we additionally need is to explicitly solve for the solution σ for x. Recall that σ is the unique solution for x, inside the unit circle, of the equation

$$x = \sum_{k \geq 0} x^k \beta_k \ . \tag{20}$$

We start by computing the β_k's in the case of Poisson arrivals. To make the analysis more insightful, we consider μ in our derivations (i.e., the service rate which by convention has an unitary value).

$$\begin{aligned}
\beta_k &= \int_0^\infty \frac{(\mu t)^k}{k!} \mathrm{e}^{-\mu t} dA(t) = \int_0^\infty \frac{(\mu t)^k}{k!} \mathrm{e}^{-\mu t} \lambda \mathrm{e}^{-\lambda t} dt \\
&= \lambda \int_0^\infty \frac{(\mu t)^k}{k!} \mathrm{e}^{-(\lambda+\mu)t} dt
\end{aligned}$$

Next, using induction and partial integration we will prove that

$$\beta_k = \frac{\lambda}{\mu} \frac{\mu^{k+1}}{(\lambda+\mu)^{k+1}} \tag{21}$$

To start the induction we consider $k = 0$, in which case we have

$$\beta_0 = \lambda \int_0^\infty \mathrm{e}^{-(\lambda+\mu)t} \ = -\frac{\lambda}{\lambda+\mu} \mathrm{e}^{-(\lambda+\mu)t} \Big|_{t=0}^{t=\infty} = \frac{\lambda}{\lambda+\mu} \ .$$

Next we assume that Eq. (21) holds for $k \in \mathbb{N}$ and we prove that it also holds for $k+1$. We see that

$$\begin{aligned}
\beta_{k+1} &= \lambda \int_0^\infty \frac{(\mu t)^{k+1}}{(k+1)!} \mathrm{e}^{-(\lambda+\mu)t} dt \\
&= \lambda \mu^{k+1} \left[-\frac{t^{k+1}}{k!} \frac{1}{\lambda+\mu} \mathrm{e}^{-(\lambda+\mu)t} \Big|_{t=0}^{t=\infty} \right. \\
&\qquad \left. + \int_{t=0}^\infty \frac{t^k}{k!} \frac{1}{\lambda+\mu} \mathrm{e}^{-(\lambda+\mu)t} dt \right] \\
&= \frac{\mu}{\lambda+\mu} \lambda \int_{t=0}^\infty \frac{(\mu t)^k}{k!} \mathrm{e}^{-(\lambda+\mu)t} dt \\
&= \frac{\mu}{\lambda+\mu} \frac{\lambda}{\mu} \frac{\mu^{k+1}}{(\lambda+\mu)^{k+1}} \\
&= \frac{\lambda}{\mu} \frac{\mu^{k+2}}{(\lambda+\mu)^{k+2}} \ .
\end{aligned} \tag{22}$$

Eq. (22) follows from the induction hypothesis, and the induction is thus complete.

Next we solve Eq. (20). For convenience, we first express β_k in terms of the traffic intensity ρ, i.e.,

$$\beta_k = \frac{\lambda}{\mu} \frac{\mu^{k+1}}{(\lambda+\mu)^{k+1}} = \rho \frac{1}{(\rho+1)^{k+1}} \ .$$

Next we can write

$$\begin{aligned}
x &= \sum_{k \geq 0} x^k \beta_k = \sum_{k \geq 0} x^k \rho \frac{1}{(\rho+1)^{k+1}} \\
&= \sum_{k \geq 0} \left(\frac{x}{1+\rho} \right)^k \frac{\rho}{1+\rho} = \frac{\rho}{1+\rho} \sum_{k \geq 0} \left(\frac{x}{1+\rho} \right)^k \\
&= \frac{\rho}{1+\rho} \frac{1}{1 - \frac{x}{1+\rho}}, |x| < 1+\rho \\
&= \frac{\rho}{1+\rho-x} \ .
\end{aligned}$$

The two solutions of the quadratic equation in x are 1 and ρ, of which $x = \rho$ is the non-trivial one. The proof is thus complete. □