Persistent-Idle Load-Distribution

Rami Atar$^{(1)}$, Isaac Keslassy$^{(1,2)}$, Gal Mendelson$^{(1)}$, Ariel Orda$^{(1)}$, Shay Vargaftik$^{(1,2)}$
1. Viterbi Faculty of Electrical Engineering, Technion.
2. VMware Research.
{atar@ee,isaac@ee,galmen@campus,ariel@ee}.technion.ac.il
shayv@vmware.com

A parallel server system is considered in which a dispatcher routes incoming jobs to a fixed number of heterogeneous servers, each with its own queue. Much effort has been previously made to design policies that use limited state information (*e.g.,* the queue lengths in a small subset of the set of servers, or the identity of the idle servers). However, existing policies either do not achieve the stability region or perform poorly in terms of job completion time. We introduce Persistent-Idle (PI), a new, perhaps counter-intuitive, load-distribution policy that is designed to work with limited state information. Roughly speaking, PI always routes to the server that has last been idle. Our main result is that this policy achieves the stability region. Since it operates quite differently from existing policies, our proof method differs from standard arguments in the literature. Specifically, large time properties of reflected random walk, along with a careful choice of a Lyapunov function, are combined to obtain a Lyapunov condition over sufficiently long time intervals. We also provide simulation results that indicate that job completion times under PI are low for different choices of system parameters, compared to several state-of-the-art load-distribution schemes.

Key words: Persistent-Idle, load-distribution, load-balancing, heterogeneous systems, parallel-server-model, Lyapunov function, state dependent drift

1. Introduction

This paper is concerned with the problem of distributing incoming jobs among a fixed set of K heterogeneous servers working in parallel, each with a dedicated buffer in which a queue can form. Jobs arrive at a dispatcher, which routes them immediately to one of the queues. Many load distribution schemes have been proposed for this setting, differing from one another in the assumptions on the information available to the dispatcher upon arrival, and the way in which this information is used. It is well known, for example, that if job processing times are available upon arrival, routing each job to the queue with the least amount of workload (the time it will take the server to complete its currently assigned jobs) results in excellent performance in terms of job completion time (*e.g.,* Foss (1982), Daley (1987), Wolff (1987), Koole (1992)). This policy is referred to as join-the-least-workload (JLW). A similar policy, also with appealing performance guarantees (*e.g.,* Winston (1977), Weber (1978), Foschini and Salz (1978)), is the well-known join-the-shortest-queue (JSQ), which uses full knowledge of current queue lengths to route the job to the shortest queue. Unfortunately, deriving such complete state information upon each arrival is often impossible in applications. Indeed, processing times are often not known in advance, and therefore workload is not observable (*e.g.,* Georgiadis et al. (2006)). Another issue is that queue length information may incur excessive communication overhead and delay involved in probing the servers for this information (*e.g.,* Lu et al. (2011)).

Several policies that use just a small amount of state information, attempting to approximate JSQ, have been proposed. For example, in the power-of-d-choices policy, denoted SQ(d), the dispatcher is only aware of the current queue lengths in $d>1$ queues chosen uniformly at random, and routes the job to the shortest among these queues. Even using d as small as 2 has been proven to result in dramatic performance improvement as compared to sending jobs randomly (*e.g.,* Vvedenskaya et al. (1996), Mitzenmacher (2001)). However, it is well known that, when servers are heterogeneous, SQ(d) is not stable (Foss and Chernova (1998)). Here and throughout the paper, we refer to a policy as *stable* if the underlying Markov chain describing the state of the system (*e.g.,* queue lengths) is positive recurrent whenever the system is sub-critical. The instability of SQ(d) is the result of the fact that all servers are equally likely to be members of the chosen d. Thus slow servers receive more work than they can handle when the overall load is high enough.

Another policy that operates under little information transmission is the power-of-memory policy, denoted SQ(d,m) (Shah and Prabhakar (2002)). Upon a job arrival, the dispatcher samples the m shortest queues from the previous round in addition to $d \geq m \geq 1$ new randomly-chosen queues. The job is routed to the shortest among these $d+m$ queues. Thus the dispatcher state information consists of $d+m$ queue lengths. It was shown that SQ(1,1) is stable (Shah and Prabhakar (2002)). However, under this policy, when a server becomes idle, it may take time for the dispatcher to get an update on this status, due to the random sampling mechanism, whereas under PI an update on idleness is immediate. This is expected to have a consequence on the cumulative idle times of servers, and as a result, on job completion times. In our simulations, we indeed observe that SQ(1,1) performs poorly in terms of job completion time compared to PI as well as other policies we examine (see §6 for details).

A recently proposed policy that is also aimed at working with reduced state information is join-the-idle-queue (JIQ) (Lu et al. (2011)). In this policy, the dispatcher maintains a list of the servers that are currently idle. When a job arrives, a member of the list is chosen (*e.g.,* the first member), and the job is routed to it. If no servers are idle, the job is routed to one out of the K servers, which is chosen uniformly at random. JIQ was analyzed in the large system limit (*e.g.,* Lu et al. (2011), and Stolyar (2015)) and was shown to achieve excellent performance for homogeneous systems at scale. However, this policy is not stable in heterogeneous systems with a fixed number of servers due to the randomization in the case where no servers are idle. This is demonstrated in Example 3 in the Appendix, and supported by the simulation results in §6.

Persistent-Idle (PI) load-distribution. The policy proposed in this paper adopts the approach of JIQ, according to which the dispatcher knows which servers are idle. In addition, the dispatcher remembers the identity of the last server it sent a job to. As in JIQ, when a job arrives, and there are idle servers, it is sent to one of them (*e.g.,* chosen uniformly at random). However, in PI, if there are no idle servers, then it is sent to the last server to which a job was sent. Thus, as long as servers are not idle, PI sends all incoming work to a single server, until a new server becomes idle, in which case all new jobs are routed to the new server, and so on.

This perhaps counter-intuitive approach is fundamentally different from the policies discussed above, which strive for *instant* balance of the load across the servers. Accordingly, this raises immediate concerns regarding the duration of the period of time during which a single queue receives all incoming work, and how this affects stability and job completion time.

Our main contribution is in establishing the stability of PI in heterogeneous systems. Due to the unique nature of PI, which does not myopically stabilize the (multidimensional) queue length process (and in which jobs can even be sent to the longest queue), standard techniques of proving stability fail. We expand on the difficulty of this problem and the motivation for our proof technique in § 3.

We also provide simulation results comparing PI and all of the policies discussed above in heterogeneous systems. The simulations indicate that PI significantly outperforms the other reduced-state policies (*i.e.,* SQ(2), SQ(1,1) and JIQ) in terms of job completion time and average workload.

Finally, similarly to token-based policies (*e.g.,* Stolyar (2015)), PI can be implemented using a fixed set of K *tokens*, each going back and forth between the dispatcher and the server to which it belongs. Namely, a server sends its token when it becomes idle; the dispatcher sends it back only with a job to process. The required communication overhead is at most one message per job (Stolyar (2015)). We also demonstrate this via simulations in §6.

Organization. In §2, the model is described, and the main result of stability is stated. §3 provides a motivation for the proof, which is then presented in §4. §5 presents and establishes several auxiliary lemmas and their proofs that are used in §4. Finally, §6 presents a simulation study that compares the performance of the PI policy with that of several load distribution policies.

2. Model and main result

We start with the introduction of some notation that is used throughout the paper.

Notation. For $a, b \in \mathbb{R}$, the maximum (resp., minimum) is denoted by $a \vee b$ (resp., $a \wedge b$), and $a^+ = a \vee 0$. $\mathbb{N}$ and $\mathbb{Z}_+$ denote the sets of positive and, respectively, nonnegative integers. For $K \in \mathbb{N}$, the set $\{1, ..., K\}$ is denoted by $[K]$.

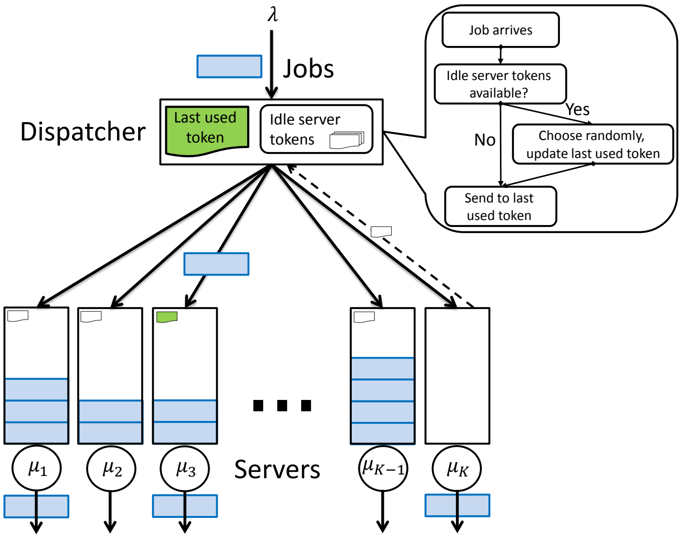


Figure 1 **A system of K parallel heterogeneous servers and a single dispatcher. The dispatcher routes incoming jobs using the Persistent-Idle load-distribution policy.**

Model. We consider a parallel-server system with a dispatcher and a set $[K]$ of heterogeneous servers, as depicted in Figure 1. Each server is work-conserving and has a buffer of infinite size in which a queue can form. The system evolves in discrete time $n \in \mathbb{N}$, such that at each time, first a possible arrival occurs and then service is given (hence a server can work on a job at the time at which it arrives). The resulting state of the queue is then determined. In case a queue is empty, we refer to the corresponding server as being *idle*. Thus the term idle refers to the state at which a server presently has no jobs to process.

Idle tokens. An *idle token* (*token* for short) is associated with each server. Each of the K tokens can be held by either the corresponding server or the dispatcher. A mechanism that assures that at all times a token is held by the dispatcher if and only if the corresponding server is idle operates as follows. The dispatcher initially holds the tokens of the servers that are idle. When an idle server becomes busy, it receives its token from the dispatcher along with the arriving job. When a non-idle server becomes idle, it sends its token back to the dispatcher. All propagation times of jobs and tokens are zero.

Persistent-Idle policy. At each time, the dispatcher selects a server to which any (potential) arrival is to be routed. The dispatcher remembers its last decision and updates it according to the following rule: if it has no tokens that are *different* from the token corresponding to its last decision, it maintains its decision; otherwise, it selects a server corresponding to one of the tokens that it holds, uniformly at random.

For ease of exposition we assume that the dispatcher may update its selection at each time regardless of whether a job arrived or not. This may appear wasteful. However, the analysis remains the same when the dispatcher may only update its selection whenever an actual arrival occurs.

Arrivals and services. Let a probability space $(\Omega,\mathcal{F},\mathbb{P})$ be given, on which all the random variables and stochastic processes to be presented are defined. Denote by $\mathbb{E}$ the expectation w.r.t. $\mathbb{P}$. Let a sequence of i.i.d. Bernoulli r.v.s $\{a_n\}$ be given, with $\mathbb{P}(a_1{=}1){=}1{-}\mathbb{P}(a_1{=}0){=}\lambda{<}1$. The event $\{a_n{=}1\}$ indicates a new arrival at time n. Let parameters $\{\mu^{(k)}\}_{k\in[K]}$, $\{\sigma^{(k)}\}_{k\in[K]}$ and sequences of i.i.d. r.v.s $\{b_n^{(k)}\}_{k\in[K]}$, taking values in $\mathbb{N}$, be given, with $\mathbb{E}[b_1^{(k)}]=1/\mu^{(k)}$, and $\mathrm{Var}(b_1^{(k)})=(\sigma^{(k)})^2<\infty$. We assume that on the event $\{a_n=1\}$, if the arriving job is assigned to server k, the amount of time it takes this server to complete the job, that we also refer to as the amount of *work* associated with it, is given by $b_n^{(k)}$. Thus, the parameters $\{\mu^{(k)}\}_{k\in K}$ represent the mean service rates of each of the servers. Since $b_n^{(k)}$ takes values in $\mathbb{N}$, we have $\mu^{(k)}\le 1$ for all k. The sequences $\{a_n\}$, $\{b_n^{(1)}\}$, $\dots$, $\{b_n^{(K)}\}$ are assumed to be mutually independent.

We do not make any assumptions on the service policy within each buffer. For example, the server can provide service to-completion according to some rule such as FIFO or LIFO, or process a unit of work belonging to each of the present jobs in a round-robin fashion.

The same mathematical model describes a more general situation, in which multiple jobs may arrive at each time, all routed together to the same server according to the aforementioned policy. The event $\{a_n{=}1\}$ is then interpreted to mean that *at least* one job arrives at time n, and $b_n^{(k)}$ models the total amount of work associated with these arrivals, provided that they are routed to server k. Whereas the results to be presented apply to this extended model, it is more convenient to stick to the original model, as far as the verbal description is concerned.

Routing. Denote by $W_n^{(k)}$ the workload in buffer k at time $n\ge 0$, defined as the total work present in the buffer after possible arrival and service at time n, and let $W_n=(W_n^{(k)})_k$. The routing is encoded in a process that takes values in $[K]$, denoted by R_n. If a job arrives at time n, it joins the queue of the server specified by the value of R_n. Recall that under the PI policy R_{n+1} may differ from R_n if and only if at time n, after arrival and service, the dispatcher holds at least one token that is *different* from R_n. We refer to such a time n as a *New Token Available* (NTA) time. Thus, if n_1 and n_2 are two consecutive NTA times, R_{n_1+1} is chosen uniformly at random from the set of available tokens at time n_1 and the routing process R_n is unchanged during $n\in(n_1,n_2]$. Define

$$\eta_n=\mathbb{1}_{\{\exists k\neq R_n:\ W_n^{(k)}=0\}},\qquad n\ge 0. \tag{1}$$

Then n is an NTA time if and only if $\eta_n{=}1$. Note that the event $\{\eta_n{=}0\}$ describes two possible scenarios: (a) there are no idle servers at time n, and (b) there is a single idle server at time n, corresponding to the last used token.

Order of events. For clarity, before turning to the Markov chain formulation, we state the order of events at each time. At time n: (1) R_n is determined with respect to W_{n-1}; (2) there is a possible arrival which is routed according to R_n; (3) service by each non-idle server is given; (4) the state of the queues, *i.e.,* W_n, is updated.

Markov chain formulation. The state of the system is given by the process $X_n = (R_n, W_n)$, which takes values in $\hat{\mathcal{S}} := [K] \times \mathbb{Z}_+^K$. This state contains two pieces of information: the routing R_n and the workload W_n at the different servers. This process satisfies the recursion, for $n \geq 1$,

$$\begin{aligned} R_n &= \begin{cases} R_{n-1}, & \eta_{n-1} = 0 \\ \xi_n, & \eta_{n-1} = 1, \end{cases} \\ W_n^{(k)} &= \left[W_{n-1}^{(k)} + a_n b_n^{(k)} \mathbb{1}_{\{R_n = k\}} - 1\right]^+, \; k \in [K], \end{aligned} \tag{2}$$

where ξ_n is a r.v. that is independent of (a_n, b_n), and whose conditional distribution, given the past states $(X_0, X_1, \ldots, X_{n-1})$, is uniform on the set $\{k: \; W_{n-1}^{(k)} = 0\}$ on the event $\{\eta_{n-1}{=}1\}$, and takes an arbitrary value on the event $\{\eta_{n-1} = 0\}$.

We define the set $\mathcal{S} \subset \hat{\mathcal{S}}$ as the collection of all states $s \in \hat{\mathcal{S}}$ such that s can be reached with positive probability from any of the states of the form $(k, 0)$, $k \in [K]$, corresponding to an empty system. The initial condition $X_0 = (R_0, W_0)$ is assumed to lie in $\mathcal{S}$ with probability 1. As a consequence, we may and will consider X_n as a Markov chain on the state space $\mathcal{S}$. Using (2), the definition of $\mathcal{S}$, and the fact that the process can reach $(k, 0)$ from $(k', 0)$ for any k, k' (thanks to the assumption $\lambda < 1$), it follows that X_n is an irreducible, time homogeneous Markov chain.

Remark 1. The information accessible to the policy consists of the set of servers that are currently idle and the last server a job was sent to. Whereas the workload at the different buffers is required in order to describe the system dynamics by means of a Markov chain, this information is not available to the algorithm. In other words, workload, queue length and individual job size information are all used in analyzing the algorithm, but are not accessible to the policy.

Our main result is the following.

Theorem 1. *Assume* $\lambda < 1 \wedge \sum_{k=1}^K \mu^{(k)}$. *Then:*

(i) X_n *is positive recurrent. Consequently,*

(ii) X_n *has a unique stationary distribution, denoted by* π_X*, and*

(iii) For any initial state $s \in \mathcal{S}$ *and any* $B \subset \mathcal{S}$, $\mathbb{P}_s(X_n \in B) \to \pi_X(B)$ *as* $n \to \infty$.

Recall that the model is defined in such a way that one always has $\lambda \leq 1$. In the case when $\sum_{k=1}^K \mu^{(k)} \leq 1$, our result states that the policy is stable. When $\sum_{k=1}^K \mu^{(k)} > 1$, the above is still the best possible result, because the limitation that λ cannot lie in the interval $(1, \sum \mu^{(k)})$ is due to the model rather than the policy.

3. Proof motivation

A standard approach to proving stability for irreducible Markov processes in general, and for queueing models in particular, relies on the Lyapunov function technique, where a function $\mathcal{L}$ is constructed, mapping the (countable) state space to $\mathbb{R}_+$, possessing a negative drift

$$\mathbb{E}[\mathcal{L}(X_1)|X_0 = x] - \mathcal{L}(x) < -\varepsilon < 0, \tag{3}$$

bounded away from zero at all states x outside of some finite set $F \subset \mathcal{S}$, and a finite expectation at all states $x \in F$, namely $\mathbb{E}[\mathcal{L}(X_1)|X_0 = x] < \infty$. The existence of such a function ensures several stability properties, such as positive recurrence and convergence to a unique invariant measure (Lemma I.3.10 of Asmussen (2008)).

The stability proofs of load-balancing policies such as JSQ (Tassiulas and Ephremides (1992), Georgiadis et al. (2006), SQ(d,m) Shah and Prabhakar (2002), Shah (2017), JIQ Lu et al. (2011)) and random routing (see Example 6.12 of (Hajek (2015))) use this technique. Specifically, they all rely on a quadratic Lyapunov function, $\mathcal{L}(x) = x^T Ax$ where A is positive semi-definite, with the state space defined as the queue length vector. Some works on stability of related models use sub-quadratic Lyapunov functions (Andrews et al. (2004), Walton (2013)).

In the case of PI, recall that the state descriptor includes the last routing decision in addition to workload, and so a Lyapunov function may, in general, depend on both components. However, the Lyapunov function that we construct does not depend on the last routing decision. We thus restrict the discussion to functions that depend only on workload or queue length.

Note that outside of any finite set there are states where the server with the largest queue receives all the incoming work with probability 1 (this corresponds to cases where the queue in the server that was idle last became the largest while none of the other queues reached zero). As a result, it is not hard to see that quadratic and sub-quadratic functions do not satisfy the negative drift condition in a single time step.

Intuitively, what makes the Lyapunov function technique effective in all the cases of the aforementioned load balancing policies is that they all strive to push the state of the system ‘towards the origin’ at every time step. As already mentioned, for PI this is not the case. There could be periods of time during which a large queue keeps receiving all incoming jobs, and this is maintained as long as all other queue lengths are non-empty. Thus the stabilizing nature of the policy (if there is any) is not to be observed in a single time step, but rather over potentially many time steps. This leads us to looking at the process at certain sampling times.

By working with workload rather than queue length we accomplish the following two goals: (1) Given a state, it is known precisely when a new server is to become idle. This is given by the minimal workload in the servers which do not receive work; (2) We do not need to make any assumptions on service time distributions or service policies within each buffer. The dynamics of the system are captured by a Markov process without the need to keep track of residual service or arrival times.

As for the sampling times, the construction that we provide uses a state-dependent number of steps (as in Malyshev and Men'shikov (1979), Meyn and Tweedie (1994)). Thus the calculation of the drift is performed with respect to time intervals that depend on the states the process traverses. The sampling times are chosen to be the NTA times, namely the times at which a token is available at the dispatcher and it is not the last used token. At these times the workload vector is always at the boundary of the positive orthant. The drift condition then only needs to be verified at these special states, rather than at the whole space. Denote the set of these states by $\mathcal{S}_b$ (where b is mnemonic for boundary). Because the Lyapunov function is observed at sampling times, the negative drift condition must be stated in terms of the duration of the intervals. That is, in place of (3), one must show

$$\mathbb{E}[\mathcal{L}(X_\tau)|X_0=x]-\mathcal{L}(x)<-\varepsilon\tau,$$

for all but finitely many x in the set $\mathcal{S}_b$, where $\varepsilon>0$ and τ denotes the next sampling time.

Whereas the dynamics of the chain are somewhat complicated, there is one element of it that can be described in relatively simple terms. Namely, between two consecutive sampling times, there is a single buffer that receives incoming traffic. During this interval, the workload at this particular buffer follows a reflected random walk starting at zero. Indeed, in this buffer, the increment of workload over a time step is given by the arriving workload minus 1, subject to the constraint that the workload remains non-negative. An additional useful fact is that the workload at all other queues decreases (or stays put) during the aforementioned interval. The significance of these observations is that they allow us to provide probabilistic estimates on the drift.

Our proposed Lyapunov function is given by

$$\mathcal{L}(x)=\sum_{k=1}^{K}\left(\mu^{(k)}x^{(k)}\right)^{1+\alpha} \tag{4}$$

where $\alpha\in(0,1)$. Before we present the proof in the next section, we show that when $\alpha=0$ or $\alpha=1$, the function $\mathcal{L}$ of (4) does not, in general, meet the desired negative drift condition.

Example 1 (Counterexample in the case α=0). Consider a system with 3 servers, each with rate $\mu<1/3$, and assume $2.5\mu<\lambda<3\mu<1$. Now, suppose that a finite set is already chosen, and consider a state outside of it with the workload vector $x_0=(x,0,0)$. Therefore, if $\alpha=0$, then $\mathcal{L}(x_0)=\mu x$. Now, the state at the next sampling time must be of the form $x_1=(x-1,0,y)$ or $x_1=(x-1,y,0)$. The server which receives the work receives on average λ/μ units of work and completes one unit if it is non-idle. Thus, roughly, the average workload of this server is equal to at least $y = \lambda/\mu - 1$, which is larger than 1.5. Thus, $\mathcal{L}(x_1) = \mu(x - 1 + y) > \mu(x + 0.5) > \mathcal{L}(x_0)$ and the drift is positive.

Example 2 (Counterexample in the case α=1). Again, consider a system with 3 servers, each with rate $\mu < 1/3$, and assume $2.5\mu < \lambda < 3\mu < 1$. Now, suppose that a finite set is already chosen, and consider a state outside of it with the workload vector $x_0 = (x, x, 0)$. Therefore, if $\alpha = 1$, then $\mathcal{L}(x_0) = 2\mu x^2$. Now, the state at the next sampling time must be of the form $x_1 = (0, 0, y)$. Every time during this interval of duration x, the third server receives on average λ/μ units of work and completes one unit if it is non-idle. Thus, roughly, the average workload of the third server is equal to at least $(\lambda/\mu - 1)x$, which is larger than $1.5x$. Thus, $\mathcal{L}(x_1) = 2.25\mu x^2 > \mathcal{L}(x_0)$ and the drift is positive.

We finally comment that the parameter α, with which the function (4) satisfies the drift condition, depends on the arrival rate λ. Hence the Lyapunov function depends on λ, which is quite uncommon in the literature of queueing systems stability.

4. Proof of main result

For $F \subset \mathcal{S}$ define the hitting time

$$\tau^F = \inf\{n \geq 1;\, X_n \in F\}.$$

Our goal is to prove that there exists a nonempty finite set $F \subset \mathcal{S}$ such that for all $s \in F$,

$$\mathbb{E}_s[\tau^F] < \infty, \tag{5}$$

where throughout the proof $\mathbb{E}_s[\cdot]$ is short for $\mathbb{E}[\cdot \mid X_0 = s]$. The irreducibility of X_n and (5) imply positive recurrence, and hence part (i) of Theorem 1, by the following Lemma:

Lemma 1 **(Lemma I.3.10, Asmussen (2008))**. *Let $\{X_n\}$ be an irreducible Markov chain and F a finite subset of its state space. Then the chain is positive recurrent provided that $\mathbb{E}_s[\tau^F] < \infty$ for all $s \in F$.*

Part (ii) then follows by I.3.6 of Asmussen (2008). Finally, by the ergodic theorem for Markov chains, Theorem I.4.2 of Asmussen (2008), part (iii) will follow once it is shown that the chain is aperiodic. To this end, it suffices to show that there exists a state with self-transition. This is clearly the case for any of the states $(k, 0)$, again owing to the assumption $\lambda < 1$.

It thus remains to show (5). Our main idea is to sample X_n at the NTA times $\{n : \eta_n = 1\}$, namely, when a token is available and it is not the last used token. To define these random times, we first define the filtration

$$\mathcal{F}_n = \sigma(X_0, \ldots, X_n, \xi_1, \ldots, \xi_{n+1}). \tag{6}$$

While including ξ_{n+1} in (6) is not mandatory, we have chosen to do so because knowing the identity of the newly used token simplifies the analysis. Using (1), define $\{N_i,\, i \geq 0\}$, an increasing sequence of $\mathcal{F}_n$ stopping times by

$$N_0 = \inf\{n \geq 0;\, \eta_n = 1\}, \tag{7}$$

and

$$N_{i+1} = \inf\{n > N_i;\, \eta_n = 1\}. \tag{8}$$

We have

Lemma 2. *The stopping times N_i are finite a.s. for all i.*

The proof is deferred to Section 5.

Define the sampled chain $Y_i = X_{N_i}$. For $F \subset \mathcal{S}$, define the stopping time

$$\sigma^F = \inf\{i \geq 1;\, Y_i \in F\}. \tag{9}$$

Thus σ^F equals the number of steps the sampled chain makes until reaching F. With a slight abuse of notation, we write $\sigma = \sigma^F$, where the corresponding set F should be clear from the context. Define the stopped sequences $\tilde{Y}_i = Y_{i\wedge\sigma}$ and $\tilde{N}_i = N_{i\wedge\sigma}$.

Denote $\Delta_i = \tilde{N}_{i+1} - \tilde{N}_i$. By (2), (7) and (8), given $\mathcal{F}_{\tilde{N}_i}$, the duration of the interval Δ_i is known. If $X_{\tilde{N}_i} \in F$, the chain has stopped and $\Delta_i = 0$. If $X_{\tilde{N}_i} \notin F$ and there are several tokens available, then $\tilde{N}_{i+1} = \tilde{N}_i + 1$. Finally, if $X_{\tilde{N}_i} \notin F$ and there is a single token available, then its corresponding server must be idle and chosen as $R_{\tilde{N}_i+1}$ while all other servers are not idle. This means that during $(\tilde{N}_i, \tilde{N}_{i+1}]$ all incoming work is routed to the server given by $R_{\tilde{N}_i+1}$. Thus Δ_i equals the number of workload units at the server with the minimal workload at time $\tilde{N}_i$, not including $R_{\tilde{N}_i+1}$ (which must be idle at time $\tilde{N}_i$).

Our main step towards proving (5) is proving that a state dependent drift criterion is fulfilled by the skeleton chain Y.

Lemma 3. *There exists $\epsilon > 0$, a function $\mathcal{L} : \mathcal{S} \to \mathbb{R}_+$ and a non-empty finite set $F \subset \mathcal{S}$, such that*

$$\mathbb{E}_s[\mathcal{L}(\tilde{Y}_{i+1}) - \mathcal{L}(\tilde{Y}_i) \mid \mathcal{F}_{\tilde{N}_i}] \leq -\epsilon\Delta_i, \quad i \geq 0,\ s \in \mathcal{S} \setminus F. \tag{10}$$

Proof. For ease of exposition, throughout this proof we drop the subscript s and write $\mathbb{E}$ instead of $\mathbb{E}_s$. We begin by fixing $\epsilon > 0$ and a function $\mathcal{L}$. We then consider a finite set F of the form

$$F = \{s = (r, w^{(1)}, ..., w^{(K)}) \in \mathcal{S} \mid \sum_k w_k \leq \mathcal{C}\}, \tag{11}$$

and show that if $\mathcal{C}$ is large enough, (10) holds. Denote $\mu_{min} = \min_k \mu^{(k)}$. Let $\epsilon_0 = \sum_{k=1}^K \mu^{(k)} - \lambda > 0$, and

$$\epsilon = (\epsilon_0 \wedge \mu_{min})/4.$$

Fix $\alpha \in (0,1]$ such that

$$\max_{r\in[K]} \big\{([\lambda - \mu^{(r)}]^+)^{1+\alpha} - \sum_{k\neq r} (\mu^{(k)})^{1+\alpha}\big\} \leq -2\epsilon.$$

The existence of such an α is guaranteed by the following Lemma:

LEMMA 4. *Let $\mu_{min} = \min_k \mu^{(k)}$, $\epsilon_0 = \sum_{k=1}^K \mu^{(k)} - \lambda > 0$ and $\epsilon = (\epsilon_0 \wedge \mu_{min})/4$. There exists $\alpha \in (0,1]$ such that*

$$\max_{r\in[K]} \big\{([\lambda - \mu^{(r)}]^+)^{1+\alpha} - \sum_{k\neq r} (\mu^{(k)})^{1+\alpha}\big\} \leq -2\epsilon.$$

The proof is deferred to Section 5.

For $s = (r, w^{(1)}, \ldots, w^{(K)}) \in \mathcal{S}$, define

$$\mathcal{L}(s) = \sum_k (\mu^{(k)} w^{(k)})^{1+\alpha}. \tag{12}$$

Denote by $r_i = R_{\tilde{N}_i+1}$ the routing decision at time $\tilde{N}_i + 1$. Using (12), we have

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\tilde{Y}_{i+1}) - \mathcal{L}(\tilde{Y}_i) \mid \mathcal{F}_{\tilde{N}_i}] &= \mathbb{E}\big[\sum_k \big(\mu^{(k)} W^{(k)}_{\tilde{N}_{i+1}}\big)^{1+\alpha} - \sum_k \big(\mu^{(k)} W^{(k)}_{\tilde{N}_i}\big)^{1+\alpha} \mid \mathcal{F}_{\tilde{N}_i}\big] \\
&= \mathbb{E}\big[\sum_{k\neq r_i} \big(\mu^{(k)}\big)^{1+\alpha} \big(\big(W^{(k)}_{\tilde{N}_{i+1}}\big)^{1+\alpha} - \big(W^{(k)}_{\tilde{N}_i}\big)^{1+\alpha}\big) \mid \mathcal{F}_{\tilde{N}_i}\big] \\
&\quad + \mathbb{E}\big[\big(\mu^{(r_i)}\big)^{1+\alpha} \big(\big(W^{(r_i)}_{\tilde{N}_{i+1}}\big)^{1+\alpha} - \big(W^{(r_i)}_{\tilde{N}_i}\big)^{1+\alpha}\big) \mid \mathcal{F}_{\tilde{N}_i}\big]
\end{aligned} \tag{13}$$

To bound the members of the right hand side of (13) we analyze three disjoint events, whose union is Ω:

$E_1 = \{i < \sigma, |\{k : W^{(k)}_{\tilde{N}_i} = 0\}| = 1\}$, $E_2 = \{i < \sigma, |\{k : W^{(k)}_{\tilde{N}_i} = 0\}| > 1\}$ and $E_3 = \{i \geq \sigma\}$.

By (2), (6) and (9), all three events are measurable with respect to $\mathcal{F}_{\tilde{N}_i}$. We prove that

$$\mathbb{E}[\big(\mathcal{L}(\tilde{Y}_{i+1}) - \mathcal{L}(\tilde{Y}_i)\big)\mathbb{1}_{E_j} \mid \mathcal{F}_{\tilde{N}_i}] \leq -\epsilon\Delta_i \mathbb{1}_{E_j},$$

where $j \in \{1,2,3\}$, thus proving (10).

The reason for analyzing E_1 and E_2 separately is that when considering two consecutive sampling times n_1 and n_2, the drift behavior of the system under the events E_1 or E_2 at n_1 is qualitatively different.

Specifically, under E_1, during the interval $(n_1,\, n_2]$ there is a single server i which is idle at time n_1 and receives all incoming work, while all other servers are non-idle. As a result, the drift during

$(n_1,\, n_2]$ is determined by the weight the Lyapunov function gives to the reflected random walk at server i as opposed to the decrease in all other servers. As discussed in Section 3, since the workload at server i at time n_2 may be the maximal workload in the system, the Lyapunov function may need to be sub-quadratic, *i.e.,* $\alpha < 1$, depending on λ.

On the other hand, under E_2, there are more idle servers at n_1 and the next sampling time is $n_2 = n_1 + 1$. The drift in this case is determined by the weight the Lyapunov function gives to the possible arrival to an idle server as opposed to the decrease in other non-idle servers in the system. As discussed in Section 3, since there are more idle servers, if λ is sufficiently large, the expected amount of work that enters the system may be larger than the work completed by the non-idle servers. This suggests using a super-linear Lyapunov function, *i.e.,* $\alpha > 0$.

On the event E_1.

In this case, since $i < \sigma$, we have $Y_i \notin F$. Additionally, since $|\{k:\ W^{(k)}_{\tilde{N}_i} = 0\}| = 1$ and $\tilde{N}_i$ is a sampling time, there is a single idle server at $\tilde{N}_i$, which is not $R_{\tilde{N}_i}$ and must be chosen as r_i. Therefore, during the time period $[\tilde{N}_i, \tilde{N}_{i+1})$, all servers except r_i are non-idle. Hence, by (2) and (8), for $k \neq r_i$, we have

$$W^{(k)}_{\tilde{N}_{i+1}} = W^{(k)}_{\tilde{N}_i} - \Delta_i. \tag{14}$$

Since ξ_{n+1} is included in $\mathcal{F}_n$, r_i is known given $\mathcal{F}_{\tilde{N}_i}$. Thus, using (14) for $k \neq r_i$,

$$\mathbb{E}\big[\big(W^{(k)}_{\tilde{N}_{i+1}}\big)^{1+\alpha}\mathbb{1}_{E_1} \,|\, \mathcal{F}_{\tilde{N}_i}\big] = \big(W^{(k)}_{\tilde{N}_i} - \Delta_i\big)^{1+\alpha}\mathbb{1}_{E_1}. \tag{15}$$

To bound the right hand side of (15) we use the following Lemma:

Lemma 5. *Fix $\alpha > 0$ and let $\delta(\alpha) = 1 - 2^{-\alpha}$. Let $x_1, x_2 \in \mathbb{Z}_+$ such that $x_1 \geq x_2$. Then:*

$$(x_1 - x_2)^{1+\alpha} \leq x_1^{1+\alpha} - x_2^{1+\alpha} - \delta(\alpha)\cdot x_1^{\alpha}\cdot \mathbb{1}_{\{x_1 > x_2 > 0\}}. \tag{16}$$

The proof is deferred to Section 5.

Using Lemma 5 in (15), for $\alpha > 0$, there exists $\delta(\alpha) > 0$ such that

$$\big(W^{(k)}_{\tilde{N}_i} - \Delta_i\big)^{1+\alpha} \leq \big(W^{(k)}_{\tilde{N}_i}\big)^{1+\alpha} - {\Delta_i}^{1+\alpha} - \delta(\alpha)\big(W^{(k)}_{\tilde{N}_i}\big)^{\alpha}\mathbb{1}_{\big\{W^{(k)}_{\tilde{N}_i} > \Delta_i > 0\big\}}. \tag{17}$$

Using (15) and (17) we obtain

$$\begin{aligned}
&\mathbb{E}\Big[\sum_{k\neq r_i}\big(\mu^{(k)}\big)^{1+\alpha}\Big(\big(W^{(k)}_{\tilde{N}_{i+1}}\big)^{1+\alpha} - \big(W^{(k)}_{\tilde{N}_i}\big)^{1+\alpha}\Big)\mathbb{1}_{E_1} \,|\, \mathcal{F}_{\tilde{N}_i}\Big] \\
&\leq \sum_{k\neq r_i}\big(\mu^{(k)}\big)^{1+\alpha}\Big(\big(W^{(k)}_{\tilde{N}_i}\big)^{1+\alpha} - \Delta_i^{1+\alpha} - \delta(\alpha)\big(W^{(k)}_{\tilde{N}_i}\big)^{\alpha}\mathbb{1}_{\big\{W^{(k)}_{\tilde{N}_i} > \Delta_i > 0\big\}} - \big(W^{(k)}_{\tilde{N}_i}\big)^{1+\alpha}\Big)\mathbb{1}_{E_1} \\
&= -\sum_{k\neq r_i}\big(\mu^{(k)}\big)^{1+\alpha}\Big(\Delta_i^{1+\alpha} + \delta(\alpha)\big(W^{(k)}_{\tilde{N}_i}\big)^{\alpha}\mathbb{1}_{\big\{W^{(k)}_{\tilde{N}_i} > \Delta_i > 0\big\}}\Big)\mathbb{1}_{E_1}.
\end{aligned} \tag{18}$$

We now bound the second term on the right hand side of (13). The server r_i is idle at time $\tilde{N}_i$ and receives all incoming work during $(\tilde{N}_i, \tilde{N}_{i+1}]$. During this time period, by (2), the workload dynamics at r_i are given by a random walk reflected at zero, with an average step size $\beta^{(r_i)} = (\lambda/\mu^{(r_i)} - 1)$. We have

Lemma 6. *For $\alpha \in (0,1]$,*

$$\mathbb{E}\big[\big(W^{(r_i)}_{\tilde{N}_{i+1}}\big)^{1+\alpha}\mathbb{1}_{E_1} \mid \mathcal{F}_{\tilde{N}_i}\big] \le \Big(((\beta^{(r_i)})^+)^{1+\alpha}{\Delta_i}^{1+\alpha} + C{\Delta_i}^{(3/4)(1+\alpha)}\Big)\mathbb{1}_{E_1}, \tag{19}$$

where $C = \max_k\{\big(16D^{(k)} + 8(D^{(k)})^{1/2}(\beta^{(k)})^+\big)^{(1+\alpha)/2}\}$, and $D^{(k)} = \lambda(\sigma^{(k)})^2 + \lambda(1-\lambda)/(\mu^{(k)})^2$.

The proof is deferred to Section 5.

Using (19), and the fact that $W^{(r_i)}_{\tilde{N}_i} = 0$, we have

$$\begin{aligned}&\mathbb{E}\big[\big(\mu^{(r_i)}\big)^{1+\alpha}\big(\big(W^{(r_i)}_{\tilde{N}_{i+1}}\big)^{1+\alpha} - \big(W^{(r_i)}_{\tilde{N}_i}\big)^{1+\alpha}\big)\mathbb{1}_{E_1} \mid \mathcal{F}_{\tilde{N}_i}\big]\\ &\le \big(\mu^{(r_i)}\big)^{1+\alpha}\Big(((\beta^{(r_i)})^+)^{1+\alpha}{\Delta_i}^{1+\alpha} + C{\Delta_i}^{(3/4)(1+\alpha)}\Big)\mathbb{1}_{E_1}.\end{aligned} \tag{20}$$

Combining (13), (18) and (20),

$$\begin{aligned}&\mathbb{E}[\big(\mathcal{L}(\tilde{Y}_{i+1}) - \mathcal{L}(\tilde{Y}_i)\big)\mathbb{1}_{E_1} \mid \mathcal{F}_{\tilde{N}_i}]\\ &\le \Big(-\sum_{k\neq r_i}\big(\mu^{(k)}\big)^{1+\alpha}\big(\Delta_i^{1+\alpha} + \delta(\alpha)\big(W^{(k)}_{\tilde{N}_i}\big)^{\alpha}\mathbb{1}_{\big\{W^{(k)}_{\tilde{N}_i} > \Delta_i > 0\big\}}\big) + \big(\mu^{(r_i)}\big)^{1+\alpha}\big(((\beta^{(r_i)})^+)^{1+\alpha}{\Delta_i}^{1+\alpha} + C{\Delta_i}^{(3/4)(1+\alpha)}\big)\Big)\mathbb{1}_{E_1}\\ &= \Big(\big(\big(\mu^{(r_i)}\big)^{1+\alpha}((\beta^{(r_i)})^+)^{1+\alpha} - \sum_{k\neq r_i}\big(\mu^{(k)}\big)^{1+\alpha}\big)\Delta_i^{1+\alpha}\\ &\quad - \sum_{k\neq r_i}\delta(\alpha)\big(\mu^{(k)}\big)^{1+\alpha}\big(W^{(k)}_{\tilde{N}_i}\big)^{\alpha}\mathbb{1}_{\big\{W^{(k)}_{\tilde{N}_i} > \Delta_i > 0\big\}} + C\big(\mu^{(r_i)}\big)^{1+\alpha}{\Delta_i}^{(3/4)(1+\alpha)}\Big)\mathbb{1}_{E_1}.\end{aligned} \tag{21}$$

The coefficient of $\Delta_i^{1+\alpha}$ satisfies

$$\big(\mu^{(r_i)}\big)^{1+\alpha}((\beta^{(r_i)})^+)^{1+\alpha} - \sum_{k\neq r_i}\big(\mu^{(k)}\big)^{1+\alpha} = ([\lambda - \mu^{(r_i)}]^+)^{1+\alpha} - \sum_{k\neq r_i}\big(\mu^{(k)}\big)^{1+\alpha} \le -2\epsilon, \tag{22}$$

where the last inequality is due to (4).

Note that the service rate weights in the Lyapunov function (12) are needed to make sure that the right hand side of (22) is negative regardless of the value of r_i.

Denote

$$\Phi_i = -\epsilon\Delta_i^{1+\alpha} - \sum_{k\neq r_i}\delta(\alpha)\big(\mu^{(k)}\big)^{1+\alpha}\big(W^{(k)}_{\tilde{N}_i}\big)^{\alpha}\mathbb{1}_{\big\{W^{(k)}_{\tilde{N}_i} > \Delta_i > 0\big\}} + C\big(\mu^{(r_i)}\big)^{1+\alpha}{\Delta_i}^{(3/4)(1+\alpha)}. \tag{23}$$

Using (21), (22) and (23) yields

$$\mathbb{E}[\big(\mathcal{L}(\tilde{Y}_{i+1}) - \mathcal{L}(\tilde{Y}_i)\big)\mathbb{1}_{E_1} \mid \mathcal{F}_{\tilde{N}_i}] \le \big(-\epsilon\Delta_i^{1+\alpha} + \Phi_i\big)\mathbb{1}_{E_1} \le \big(-\epsilon\Delta_i + \Phi_i\big)\mathbb{1}_{E_1}.$$

Finally, we have

Lemma 7. *There exists $\mathcal{C}_1 > 0$ such that if $\sum_{k\in[K]} W^{(k)}_{\tilde{N}_i} > \mathcal{C}_1$, then $\Phi_i\mathbb{1}_{E_1} \le 0$.*

The proof is deferred to Section 5.

Therefore taking $\mathcal{C} \geq \mathcal{C}_1$ in (11) completes the proof of this case.

On the event E_2.

In this case, $|\{k: W^{(k)}_{\tilde{N}_i} = 0\}| > 1$, *i.e.,* the dispatcher has more than a single token available at $\tilde{N}_i$. Therefore

$$\Delta_i = \tilde{N}_{i+1} - \tilde{N}_i = 1. \tag{24}$$

Additionally, $Y_i \notin F$. We provide a condition on F which ensures that the decrease in the server with the maximal workload results in a negative average drift. Similarly to (19), since r_i is measurable with respect to $\mathcal{F}_{\tilde{N}_i}$ and $W^{(r_i)}_{\tilde{N}_i} = 0$, by (24) and Lemma 6, there exists a constant $\gamma > 0$ such that

$$\mathbb{E}\big[\big(\mu^{(r_i)}\big)^{1+\alpha}\big(W^{(r_i)}_{\tilde{N}_{i+1}}\big)^{1+\alpha}\mathbb{1}_{E_2} \mid \mathcal{F}_{\tilde{N}_i}\big] \leq \gamma \mathbb{1}_{E_2}. \tag{25}$$

Denote $C_{min} = \min_k \big(\mu^{(k)}\big)^{1+\alpha}$, and let

$$\mathcal{C}_2 = K\big((\gamma+\epsilon)/(C_{min}\delta(\alpha))\big)^{1/\alpha} \vee K. \tag{26}$$

By taking $\mathcal{C} \geq \mathcal{C}_2$ in (11), $\sum_k W^{(k)}_{\tilde{N}_i} > \mathcal{C} \geq \mathcal{C}_2$. Thus we have

$$\max_{k \neq r_i}\{W^{(k)}_{\tilde{N}_i}\} > \mathcal{C}_2/K \geq 1. \tag{27}$$

Therefore there exists a $k^* \neq r_i$ such that $W^{(k^*)}_{\tilde{N}_i} = \max_{k \neq r_i}\{W^{(k)}_{\tilde{N}_i}\}$ and $W^{(k^*)}_{\tilde{N}_{i+1}} = W^{(k^*)}_{\tilde{N}_i} - 1$. Additionally, for $k \notin \{r_i, k^*\}$, $W^{(k)}_{\tilde{N}_{i+1}} - W^{(k)}_{\tilde{N}_i} \leq 0$. Hence

$$\mathbb{E}\big[\sum_{k \neq r_i}\big(\mu^{(k)}\big)^{1+\alpha}\big(\big(W^{(k)}_{\tilde{N}_{i+1}}\big)^{1+\alpha} - \big(W^{(k)}_{\tilde{N}_i}\big)^{1+\alpha}\big)\mathbb{1}_{E_2} \mid \mathcal{F}_{\tilde{N}_i}\big] \leq C_{min}\big(\big(W^{(k^*)}_{\tilde{N}_i} - 1\big)^{1+\alpha} - \big(W^{(k^*)}_{\tilde{N}_i}\big)^{1+\alpha}\big)\mathbb{1}_{E_2}. \tag{28}$$

By (27), $W^{(k^*)}_{\tilde{N}_i} > 1$. Thus, by Lemma 5,

$$\big(W^{(k^*)}_{\tilde{N}_i} - 1\big)^{1+\alpha} \leq \big(W^{(k^*)}_{\tilde{N}_i}\big)^{1+\alpha} - 1 - \delta(\alpha)\big(W^{(k^*)}_{\tilde{N}_i}\big)^{\alpha}. \tag{29}$$

By (27) and (29),

$$\big(W^{(k^*)}_{\tilde{N}_i} - 1\big)^{1+\alpha} - \big(W^{(k^*)}_{\tilde{N}_i}\big)^{1+\alpha} \leq -\delta(\alpha)\big(W^{(k^*)}_{\tilde{N}_i}\big)^{\alpha} \leq -\delta(\alpha)\big(\mathcal{C}_2/K\big)^{\alpha}. \tag{30}$$

Using (25), (28) and (30) in (13), followed by (26) and (24), we obtain

$$\mathbb{E}[\mathcal{L}(\tilde{Y}_{i+1}) - \mathcal{L}(\tilde{Y}_i)\mathbb{1}_{E_2} \mid \mathcal{F}_{\tilde{N}_i}] \leq \big(\gamma - C_{min}\delta(\alpha)\big(\mathcal{C}_2/K\big)^{\alpha}\big)\mathbb{1}_{E_2} \leq -\epsilon\mathbb{1}_{E_2} = -\epsilon\Delta_i\mathbb{1}_{E_2}.$$

On the event E_3.

In this case $i \geq \sigma$. Therefore $\tilde{Y}_i \in F$, $\tilde{N}_{i+1} = \tilde{N}_i$, $\Delta_i = 0$ and $\tilde{Y}_{i+1} = \tilde{Y}_i$. Thus

$$\mathbb{E}\big[\big(\mathcal{L}(\tilde{Y}_{i+1}) - \mathcal{L}(\tilde{Y}_i)\big)\mathbb{1}_{E_3} \mid \mathcal{F}_{\tilde{N}_i}\big] = -\epsilon\Delta_i\mathbb{1}_{E_3} = 0.$$

Finally, fixing $\mathcal{C} = \mathcal{C}_1 \vee \mathcal{C}_2$ in (11), where $\mathcal{C}_1$ and $\mathcal{C}_2$ are given in (47) and (26) respectively, completes the proof of Lemma 3. Q.E.D.

Proof of Theorem 1. We now use Lemma 3 to prove (5). Define $\mathcal{S}^* \subset \mathcal{S}$ as the collection of states that can be reached at sampling times, namely

$$\mathcal{S}^* = \{s = (r, w^{(1)}, \ldots, w^{(K)}) \in \mathcal{S} : w^{(k)} = 0 \text{ for some } k \neq r\}.$$

Fix an initial state $s_0 = (r_0, w_0^{(1)}, \ldots, w_0^{(K)}) \in F$. Let

$$n_0 = \begin{cases} \min_{k \neq r_0}\{w_0^{(k)}\} & \text{if } s_0 \notin \mathcal{S}^* \\ w_0^{(r_0)} \vee 1 & \text{if } s_0 \in \mathcal{S}^* \end{cases},$$

We argue that conditioned on $X_0 = s_0$, n_0 is a sampling time and consequently one has $X_{n_0} \in \mathcal{S}^*$.

Case 1: $s_0 \notin \mathcal{S}^*$. The dispatcher has no tokens available which are different from r_0, and $w^{(k)} > 0$ for $k \neq r_0$. Thus the first sampling time occurs when the least loaded server becomes idle.

Case 2: $s_0 \in \mathcal{S}^*$. In this case, time 0 is a sampling time and there is a token available different from r_0. If $w^{(r_0)} = 0$, time 1 is necessarily a sampling time as well. Otherwise, r_0 receives no work until it becomes idle and $w^{(r_0)}$ is a sampling time. Note that times belonging to $(0, w^{(r_0)})$ may be sampling times as well.

Denote the n-step transition probabilities by

$$p_{s_1 s_2}^{(n)} = \mathbb{P}(X_n = s_2 \mid X_0 = s_1), \quad s_1, s_2 \in \mathcal{S}.$$

Also, $p_{s_0 s}^{(n_0)} = 0$ for $s \notin \mathcal{S}^*$. We have

$$\begin{aligned} \mathbb{E}_{s_0}[\tau^F] &= \sum_{s \in \mathcal{S}^*} p_{s_0 s}^{(n_0)} \mathbb{E}_{s_0}[\tau^F \mid X_{n_0} = s] = \sum_{s \in \mathcal{S}^* \cap F} p_{s_0 s}^{(n_0)} \mathbb{E}_{s_0}[\tau^F \mid X_{n_0} = s] + \sum_{s \in \mathcal{S}^* \setminus F} p_{s_0 s}^{(n_0)} \mathbb{E}_{s_0}[\tau^F \mid X_{n_0} = s] \\ &\leq n_0 \sum_{s \in \mathcal{S}^* \cap F} p_{s_0 s}^{(n_0)} + \sum_{s \in \mathcal{S}^* \setminus F} p_{s_0 s}^{(n_0)} \mathbb{E}_{s_0}[\tau^F \mid X_{n_0} = s] \leq n_0 + \sum_{s \in \mathcal{S}^* \setminus F} p_{s_0 s}^{(n_0)} \mathbb{E}_s[\tau^F], \end{aligned} \tag{31}$$

where the last inequality is due to X_n being a Markov chain. To upper-bound $\mathbb{E}_s[\tau^F]$, we use the fact that if $X_0 = s \in \mathcal{S}^* \setminus F$ then $\tau^F \leq N_\sigma$ a.s., where N_σ is the time it takes sampled chain Y_n to hit F. As a consequence of Lemma 3, $\sigma < \infty$ a.s., as we prove in Lemma 8. Thus N_σ is well defined. Therefore, we proceed with finding an upper-bound on $\mathbb{E}_s[N_\sigma]$. By (7) and the fact that $s \in \mathcal{S}^*$ we have

$$\tilde{Y}_0 = Y_0 = s, \quad \tilde{N}_0 = N_0 = 0. \tag{32}$$

We now apply Lemma 3. Taking expectation on both sides of (10), we obtain

$$\mathbb{E}_s[\mathcal{L}(\tilde{Y}_{i+1}) - \mathcal{L}(\tilde{Y}_i)] \leq -\epsilon \mathbb{E}_s[\tilde{N}_{i+1} - \tilde{N}_i].$$

Summing over $i \in [0, m-1]$ and using (32) yields

$$\mathbb{E}_s[\mathcal{L}(\tilde{Y}_m)] - \mathcal{L}(s) \leq -\epsilon \mathbb{E}_s[\tilde{N}_m].$$

Using the fact that $\mathbb{E}_s[\mathcal{L}(\tilde{Y}_m)] \geq 0$ and rearranging yields

$$\mathbb{E}_s[\tilde{N}_m] \leq \mathcal{L}(s)/\epsilon. \tag{33}$$

Lemma 8. *Assume the conditions of Lemma 3 hold and* $s \in S^* \setminus F$. *Then,* $\mathbb{E}_s[\sigma] < \infty$ *and consequently* $\sigma < \infty$ *a.s.*

The proof is deferred to Section 5.

Recall $\tilde{N}_i = N_{i \wedge \sigma}$. By Lemma 8, the monotone convergence theorem and (33),

$$\mathbb{E}_s[N_\sigma] = \mathbb{E}_s \lim_{i\to\infty} N_{i\wedge\sigma} = \lim_{i\to\infty} \mathbb{E}_s[N_{i\wedge\sigma}] = \lim_{i\to\infty} \mathbb{E}_s[\tilde{N}_i] \le \mathcal{L}(s)/\epsilon. \tag{34}$$

Revisiting (31), using $\mathbb{E}_s[\tau^F] \le \mathbb{E}_s[N_\sigma]$ and (34) we obtain

$$\mathbb{E}_{s_0}[\tau^F] \le n_0 + \epsilon^{-1} \sum_{s \in \mathcal{S}^* \setminus F} p_{s_0 s}^{(n_0)} \mathcal{L}(s) \le n_0 + \epsilon^{-1} \mathbb{E}_{s_0}[\mathcal{L}(X_{n_0})] < \infty,$$

where the last inequality is due to the fact that

$$\mathbb{E}_{s_0}[\mathcal{L}(X_{n_0})] \le \sum_{k\in[K]} \big(\mu^{(k)}\big)^{1+\alpha} \mathbb{E}_{s_0}\big[\big(w_0^{(k)} + \sum_{n=1}^{n_0} a_n b_n^{(k)}\big)^{1+\alpha}\big] < \infty.$$

This concludes the proof. Q.E.D.

5. Lemmas

In this section we prove several lemmas that were used in the proof of the main result. For the reader's convenience we restate these lemmas before providing their proofs.

Lemma 2 *The stopping times* N_i *are finite a.s. for all* i.

Proof. Fix $i{\ge}0$, and consider the event $\{N_i{=}\infty\}$. On this event, there exists an infinite sequence of consecutive times such that apart from one server, all other servers are busy and receive no input. Therefore the workload in these servers must have been infinite at some point in time, which contradicts the finiteness of the initial condition and the arrival process. Thus the probability of this event is zero and the claim follows.

Q.E.D.

Lemma 4 *Let* $\mu_{min} = \min_k \mu^{(k)}$, $\epsilon_0 = \sum_{k=1}^K \mu^{(k)} - \lambda > 0$ *and* $\epsilon = (\epsilon_0 \wedge \mu_{min})/4$. *There exists* $\alpha \in (0,1]$ *such that*

$$\max_{r\in[K]} \big\{([\lambda - \mu^{(r)}]^+)^{1+\alpha} - \sum_{k\ne r} (\mu^{(k)})^{1+\alpha}\big\} \le -2\epsilon.$$

Proof. Define $g_r(\alpha) = ([\lambda - \mu^{(r)}]^+)^{1+\alpha} - \sum_{k\ne r} (\mu^{(k)})^{1+\alpha}$. Then,

$$g_r(0) = [\lambda - \mu^{(r)}]^+ - \sum_{k\ne r} \mu^{(k)} = -\big(\epsilon_0 \wedge \sum_{k\ne r} \mu^{(k)}\big) \le -\big(\epsilon_0 \wedge \mu_{min}\big) = -4\epsilon. \tag{35}$$

Define $h(\alpha) = \max_{r\in[K]} g_r(\alpha)$. By (35), $h(0) \le -4\epsilon$. Since $g_r(\alpha)$ are continuous functions of α, so is $h(\alpha)$. Thus, there exists $\alpha \in (0,1]$ such that $h(\alpha) \le -2\epsilon$, which concludes the proof.

Q.E.D.

Lemma 5 *Fix $\alpha > 0$ and let $\delta(\alpha) = 1 - 2^{-\alpha}$. Let $x_1, x_2 \in \mathbb{Z}_+$ such that $x_1 \geq x_2$. Then:*

$$(x_1 - x_2)^{1+\alpha} \leq x_1^{1+\alpha} - x_2^{1+\alpha} - \delta(\alpha) \cdot x_1^{\alpha} \cdot \mathbb{1}_{\{x_1 > x_2 > 0\}}. \tag{36}$$

Proof. If $x_1{=}x_2$ or $x_2{=}0$ the claim trivially holds. Therefore, we examine $x_1{>}x_2{>}0$, *i.e.,* $x_1, x_2{\in}\mathbb{N}$ and $\mathbb{1}_{\{x_1 > x_2 > 0\}}{=}1$. Denote $x_1{=}x+k$, $x_2{=}x$ where $k{\in}\mathbb{N}$. Rearranging (36), we must prove that

$$\delta(\alpha)(x+k)^{\alpha} \leq (x+k)^{1+\alpha} - x^{1+\alpha} - k^{1+\alpha}. \tag{37}$$

Equivalently, after dividing (37) by $(x+k)^{\alpha}$, we must prove that

$$\delta(\alpha) \leq x + k - x \cdot \Big(\frac{x}{x+k}\Big)^{\alpha} - k \cdot \Big(\frac{k}{x+k}\Big)^{\alpha} = x \cdot \Big(1 - \Big(\frac{x}{x+k}\Big)^{\alpha}\Big) + k \cdot \Big(1 - \Big(\frac{k}{x+k}\Big)^{\alpha}\Big). \tag{38}$$

We observe that both $(\frac{x}{x+k})^{\alpha}{<}1$ and $(\frac{k}{x+k})^{\alpha}{<}1$. Therefore, using the fact that $x, k \in \mathbb{N}$, we obtain

$$\Big(1 - \Big(\frac{x}{x+k}\Big)^{\alpha}\Big) + \Big(1 - \Big(\frac{k}{x+k}\Big)^{\alpha}\Big) \leq x \cdot \Big(1 - \Big(\frac{x}{x+k}\Big)^{\alpha}\Big) + k \cdot \Big(1 - \Big(\frac{k}{x+k}\Big)^{\alpha}\Big). \tag{39}$$

Since $\frac{x}{x+k} + \frac{k}{x+k} = 1$, we have $\min\{\frac{x}{x+k}, \frac{k}{x+k}\} \leq \frac{1}{2}$. Thus

$$1 - 2^{-\alpha} \leq \Big(1 - \Big(\frac{x}{x+k}\Big)^{\alpha}\Big) + \Big(1 - \Big(\frac{k}{x+k}\Big)^{\alpha}\Big). \tag{40}$$

By (39) and (40), (38) follows, which concludes the proof.

Q.E.D.

Lemma 6 *Let a sequence of i.i.d. Bernoulli r.v.s $\{a_n\}$ be given, with $E[a_1] = \lambda$. Let $\mu > 0$ and a sequence of i.i.d. r.v.s $\{b_n\}$ be given, with $E[b_1] = 1/\mu$, and Var$(b_1) = \sigma^2$. Let H_n be a random walk such that $H_0 = 0$ and for $n \geq 1$,*

$$H_n = \sum_{l=1}^{n} (a_l b_l - 1).$$

Define the corresponding reflected random walk W_n by

$$W_n = H_n + \max_{0 \leq m \leq n} (-H_m).$$

Fix $\alpha \in (0, 1]$ and let $\beta = \lambda/\mu - 1$. Then

$$\mathbb{E}[W_n^{1+\alpha}] \leq (\beta^+)^{1+\alpha} n^{1+\alpha} + C n^{(3/4)(1+\alpha)},$$

where $C = (16D + 8D^{1/2}\beta^+)^{(1+\alpha)/2}$ and $D = \lambda\sigma^2 + \lambda(1-\lambda)/\mu^2$.

Proof. First, since $0 < (1+\alpha)/2 \leq 1$, by Jensen's inequality we have

$$\mathbb{E}[W_n^{1+\alpha}] = \mathbb{E}[W_n^{2(1+\alpha)/2}] \leq \Big(\mathbb{E}[W_n^2]\Big)^{(1+\alpha)/2}. \tag{41}$$

We turn to analyze $\mathbb{E}[W_n^2]$. Let $M_n = H_n - \beta n$. The process M_n is a symmetric random walk, and therefore a martingale. We have

$$\max_{0\leq m\leq n}(-H_m) = \max_{0\leq m\leq n}(-H_m + \beta m - \beta m) \leq \max_{0\leq m\leq n}(-M_m) + \max_{0\leq m\leq n}(-\beta m) \leq \max_{0\leq m\leq n}|M_m| + (-\beta)^+ n.$$

Therefore

$$\begin{aligned} W_n^2 &= (H_n + \max_{0\leq m\leq n}(-H_m))^2 \leq (M_n + \beta n + \max_{0\leq m\leq n}|M_m| + (-\beta)^+ n)^2 = (M_n + \max_{0\leq m\leq n}|M_m| + \beta^+ n)^2 \\ &\leq (2\max_{0\leq m\leq n}|M_m| + \beta^+ n)^2 = 4(\max_{0\leq m\leq n}|M_m|)^2 + 4\beta^+ n \max_{0\leq m\leq n}|M_m| + (\beta^+)^2 n^2, \end{aligned} \tag{42}$$

where in the first inequality we have used the fact that $W_n \geq 0$. By the L^p maximal inequality (Theorem 5.4.3 of Durrett (2010)),

$$\mathbb{E}[(\max_{0\leq m\leq n}|M_m|)^2] \leq 4n\mathrm{Var}(a_1 b_1) = 4n(\lambda\sigma^2 + \lambda(1-\lambda)/\mu^2) = 4nD, \tag{43}$$

and using Jensen's inequality,

$$\mathbb{E}[\max_{0\leq m\leq n}|M_m|] \leq (\mathbb{E}[(\max_{0\leq m\leq n}|M_m|)^2])^{1/2} \leq 2n^{1/2}D^{1/2}. \tag{44}$$

Using (42), (43) and (44) we obtain

$$\mathbb{E}[W_n^2] \leq 16Dn + 8D^{1/2}\beta^+ n^{3/2} + (\beta^+)^2 n^2 \leq (16D + 8D^{1/2}\beta^+)n^{3/2} + (\beta^+)^2 n^2. \tag{45}$$

By (41) and (45),

$$\mathbb{E}[W_n^{1+\alpha}] \leq \Big((16D + 8D^{1/2}\beta^+)n^{3/2} + (\beta^+)^2 n^2\Big)^{(1+\alpha)/2}.$$

Now, for $\alpha \in (0,1]$ the function $f(x) = x^{(1+\alpha)/2}$ is concave, with $f(0) = 0$. Therefore it is subadditive, namely $f(a+b) \leq f(a) + f(b)$, for $a, b \geq 0$. The result follows.

Q.E.D.

Lemma 7 *There exists $\mathcal{C}_1 > 0$ such that if $\sum_{k\in[K]} W_{\tilde{N}_i}^{(k)} > \mathcal{C}_1$, then $\Phi_i \mathbb{1}_{E_1} \leq 0$, where*

$$\Phi_i = -\epsilon\Delta_i^{1+\alpha} - \sum_{k\neq r_i} \delta(\alpha)\big(\mu^{(k)}\big)^{1+\alpha}\big(W_{\tilde{N}_i}^{(k)}\big)^{\alpha}\mathbb{1}_{\big\{W_{\tilde{N}_i}^{(k)} > \Delta_i > 0\big\}} + C\big(\mu^{(r_i)}\big)^{1+\alpha}\Delta_i^{(3/4)(1+\alpha)}.$$

Proof. On E_1, $i < \sigma$ and therefore $\Delta_i \geq 1$. Denote

$$d_1 = \min_k\{\delta(\alpha)\big(\mu^{(k)}\big)^{1+\alpha}\},$$

and

$$d_2 = \max_k\{C\big(\mu^{(k)}\big)^{1+\alpha}\}.$$

Then

$$\begin{aligned}\Phi_i &\le -\epsilon\Delta_i^{1+\alpha} - d_1 \sum_{k\neq r_i} \big(W_{\tilde{N}_i}^{(k)}\big)^{\alpha} \mathbb{1}_{\big\{W_{\tilde{N}_i}^{(k)} > \Delta_i > 0\big\}} + d_2 {\Delta_i}^{(3/4)(1+\alpha)} \\ &= {\Delta_i}^{(3/4)(1+\alpha)}\big(d_2 - \epsilon\Delta_i^{(1+\alpha)/4}\big) - d_1 \sum_{k\neq r_i} \big(W_{\tilde{N}_i}^{(k)}\big)^{\alpha} \mathbb{1}_{\big\{W_{\tilde{N}_i}^{(k)} > \Delta_i > 0\big\}}.\end{aligned} \tag{46}$$

Let $u = (d_2/\epsilon)^{4/(1+\alpha)}$, $v = K\big(u^{(3/4)(1+\alpha)} d_2/d_1\big)^{1/\alpha}$ and

$$\mathcal{C}_1 = (Ku) \vee v. \tag{47}$$

We consider two cases corresponding to the sign of $d_2 - \epsilon\Delta_i^{(1+\alpha)/4}$ in (46).

Case 1: $\Delta_i \ge u$. In this case, by the definition of u, $d_2 - \epsilon\Delta_i^{(1+\alpha)/4} \le 0$. Therefore, by (46), $\Phi_i \le 0$.

Case 2: $\Delta_i < u$. In this case, $d_2 - \epsilon\Delta_i^{(1+\alpha)/4} > 0$. First, it holds that

$${\Delta_i}^{(3/4)(1+\alpha)}\big(d_2 - \epsilon\Delta_i^{(1+\alpha)/4}\big) < u^{(3/4)(1+\alpha)} d_2. \tag{48}$$

Second, we show that

$$\sum_{k\neq r_i} \big(W_{\tilde{N}_i}^{(k)}\big)^{\alpha} \mathbb{1}_{\big\{W_{\tilde{N}_i}^{(k)} > \Delta_i > 0\big\}} \ge (\mathcal{C}_1/K)^{\alpha}. \tag{49}$$

Applying (48) and (49) in (46), we obtain

$$\Phi_i \le u^{(3/4)(1+\alpha)} d_2 - d_1 (\mathcal{C}_1/K)^{\alpha} \le 0,$$

where the last inequality is due to (47). Thus, it remains to prove (49). By (47), $u \le \mathcal{C}_1/K$. Since $\Delta_i < u$, and $\sum_{k\in[K]} W_{\tilde{N}_i}^{(k)} > \mathcal{C}_1$, we have

$$\max_{k\neq r_i}\{W_{\tilde{N}_i}^{(k)}\} \ge \mathcal{C}_1/K \ge u > \Delta_i. \tag{50}$$

Therefore, there exists a $k^* \neq r_i$ such that $W_{\tilde{N}_i}^{(k^*)} = \max_{k\neq r_i}\{W_{\tilde{N}_i}^{(k)}\}$ and $W_{\tilde{N}_i}^{(k^*)} > \Delta_i \ge 1$. Thus,

$$\sum_{k\neq r_i} \big(W_{\tilde{N}_i}^{(k)}\big)^{\alpha} \mathbb{1}_{\big\{W_{\tilde{N}_i}^{(k)} > \Delta_i > 0\big\}} \ge (W_{\tilde{N}_i}^{(k^*)})^{\alpha} \ge (\mathcal{C}_1/K)^{\alpha},$$

where the last inequality is due to (50). This concludes the proof.

Q.E.D.

Lemma 8 *Assume the conditions of Lemma 3 hold and $s \in S^* \setminus F$. Then, $\mathbb{E}_s[\sigma] < \infty$ and consequently $\sigma < \infty$ a.s.*

Proof. If $i < \sigma$ then $\Delta_i \ge 1$. Otherwise, $\Delta_i = 0$. Therefore, $\mathbb{1}_{\{i<\sigma\}} \le \Delta_i$. Using Lemma 3, we obtain

$$\mathbb{E}_s[\mathcal{L}(\tilde{Y}_{i+1}) - \mathcal{L}(\tilde{Y}_i) \mid \mathcal{F}_{\tilde{N}_i}] \le -\epsilon \mathbb{1}_{\{i<\sigma\}}.$$

Taking expectation and summing over $i \in [0, m]$ yields

$$\mathbb{E}_s[\mathcal{L}(\tilde{Y}_{m+1})] - \mathcal{L}(s) \leq -\epsilon \sum_{i=0}^{m} \mathbb{P}(\sigma > i),$$

where we have used the fact that $\tilde{Y}_0 = s$ since $s \in S^*$. Using $\mathbb{E}_s[\mathcal{L}(\tilde{Y}_{m+1})] \geq 0$, and rearranging, we have

$$\sum_{i=0}^{m} \mathbb{P}(\sigma > i) \leq \mathcal{L}(s)/\epsilon.$$

Hence

$$\mathbb{E}_s[\sigma] = \lim_{m\to\infty} \sum_{i=0}^{m} \mathbb{P}(\sigma > i) \leq \mathcal{L}(s)/\epsilon < \infty.$$

Q.E.D.

6. Simulations

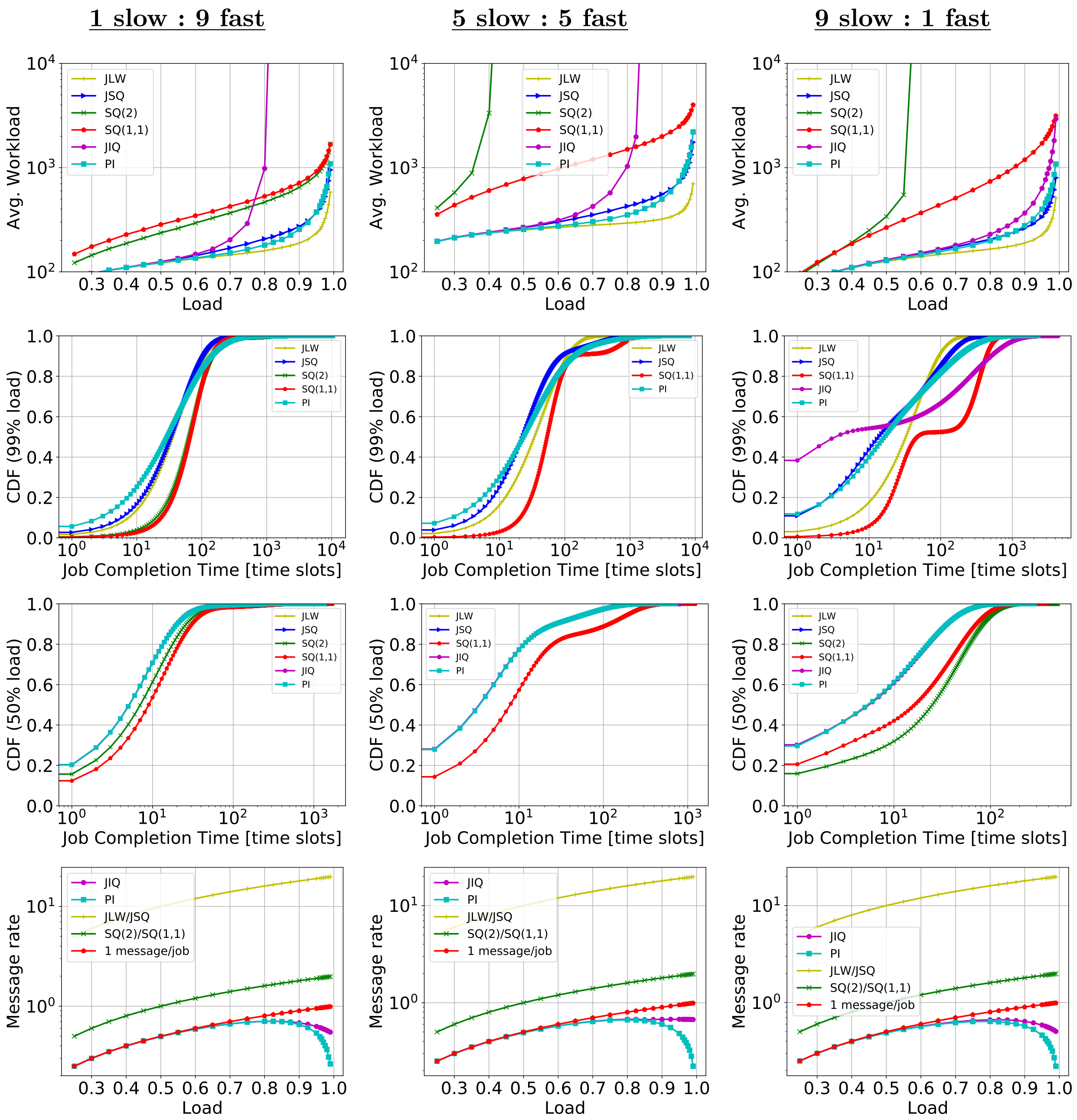
We turn to present simulation results, which compare PI's performance with that of several load-distribution policies, namely JLW, JSQ, SQ(2), SQ(1,1) and JIQ. As shall be seen, the simulations indicate that PI is indeed stable. Moreover, they consistently show that its performance is comparable to JSQ's, even though JSQ utilizes full queue-length state information. Moreover, they also indicate that JIQ and SQ(2) are not stable for heterogeneous systems and that SQ(1,1) achieves poor performance with respect to job completion time.

6.1. Simulation settings

We consider a system with 10 heterogeneous servers, some being slow servers, working at a rate μ, and the others being fast servers, working at a rate 10μ. There are three scenarios, which correspond to different partitions of the servers into the two classes. More specifically, we consider the following slow:fast ratios: 1:9, 5:5 and 9:1. Given the ratio, the exact service rates are determined such that their sum is one. For example, if the ratio is 1:9, there is one slow server and nine fast servers. Their service rates sum to $\mu + 9 \cdot 10\mu = 91\mu$, so μ is set to equal $1/91$. Thus, for $\lambda < 1$, the system is sub-critical. The service time distribution is Geometric, with the parameter depending on the server (*e.g.,* Geometric($1/\mu$) for slow servers).

For each scenario, we run simulations on a large number of time slots for different loads, where all policies receive the same input process, but possibly make different routing decisions. The number of time slots was chosen such that the difference in the outputs of different runs at the maximal load were negligible. We present the following graphs:

Average workload in steady state versus load. For each simulation run with a given load we calculate the average workload in the system (over all time-slots, after an initial duration required for convergence). Under all policies, whenever the chain is positive recurrent, it is also ergodic. Thus



1 slow : 9 fast

5 slow : 5 fast

9 slow : 1 fast

Figure 2 **Performance of different policies with 10 heterogeneous servers with different slow:fast server ratios.**

the average workload converges, and one simulation run is enough to calculate the steady-state average workload. Also, there are points that do not appear on the graphs since they are vertically truncated. However, since the truncation is at very high average workloads, these missing points indicate either poor performance or instability.

CDF of JCT in steady state for 99% load. We keep track of the job completion times of all jobs that entered the system during the simulations, with λ=0.99. We calculate the cumulative distribution function (CDF). Policies that are unstable under this load are not shown in the graph.

CDF of JCT in steady state for 50% load. This is the same as the last graph but with λ=0.5.

Message rate vs. load. For JIQ and PI, for each simulation run with a given load, we calculate the number of times a token is passed from a server to the dispatcher (the reverse path is not counted because the token is sent along with a job). We then divide the number of time slots by this number to obtain the message rate. The other policies' message rate is more straightforward to obtain. Therefore the graph contains a calculation and not a measurement. For JLW / JSQ it is just the number of servers needed to be probed (10), times the load (λ), times 2 (the dispatcher needs to probe and receive an answer). Similarly, for SQ(2) and SQ(1,1) it is $2 \cdot 2 \cdot \lambda$. The graph also contains the message rate corresponding to one-message-per-job as a baseline for comparison.

Remark 2. The number of messages needed for the above policies highly depends on the application and the implementation. For example, if a server could inform the dispatcher whenever it finishes a job, JSQ can be implemented with just one message per job. Thus we only present a comparison for the case where JLW / JSQ / SQ(2) / SQ(1,1) require probing the servers.

6.2. Simulation results

Stability and average workload. As can be seen from the average-workload-vs.-load graphs, for all the different slow:fast ratios, PI appears to achieve the stability region. Also, the average workload under PI is less than or equal to that of JSQ for most loads.

SQ(1,1) also appears to achieve the stability region, but with a much larger average workload. SQ(2) appears to be unstable for moderately-high loads for the 5:5 and 9:1 ratios. Also, whenever SQ(2) is stable, its average workload is much higher than under PI. JIQ appears unstable for loads over 80% for the 1:9 and 5:5 ratios. For all ratios, its average workload becomes larger than under PI for moderate-to-high loads.

Remark 3. The ratios for which SQ(2) and JIQ do not achieve the stability region differ, thus if the service rates are unknown, it is not clear which of these policies can be used and for what loads.

Job completion time. Under PI, for all ratios, jobs are completed faster than under SQ(1,1) and SQ(2). At 50% load, the CDFs of PI, JIQ, JLW, and JSQ coincide. At 99% load, JIQ is unstable for the first two ratios. In the third ratio, it appears to be stable; however, its performance quickly degrades as load increases. Remarkably, even at 99% load, PI achieves similar results to JLW and JSQ.

Message rate. The simulations indicate that up to high loads (around 90%), PI and JIQ have the same message rate which approximately equals the incoming job rate since there is little queue buildup. For higher loads, when there is a queue buildup, JIQ is either unstable or has a larger message rate. Compared to the remaining policies, PI's message rate is significantly smaller and keeps decreasing as the offered load increases.

Appendix. Instability of JIQ

The example below demonstrates the claim made in the introduction that JIQ is not always stable.

Example 3. Consider the following simple example in continuous time, where the arrival process and service times are deterministic. There are 2 servers, with service rates $\mu_1 = 2/11$ and $\mu_2 = 10/11$, such that a job processed in the first (respectively, second) server takes precisely 11/2 (respectively, 11/10) time units. A new job arrives upon every unit of time. Thus, the input rate is 1, and the sum of the service rates is 12/11; hence the system is sub-critical. If server 2 is idle when a job arrives, it is necessarily not idle when the next job arrives (because it requires more than a single unit of time to complete a job). Thus, *at least* 1/2 of the jobs are either randomly routed to one of the servers, or are routed to server 1 if it is idle. Therefore, on average, server 1 receives at least 1/4 of the jobs. Since $1/4 > 2/11$ server, 1 is overloaded, and the system is unstable.

Acknowledgment. The authors are grateful to two referees for careful reading of the manuscript and valuable comments. RA is supported in part by the ISF (grant 1184/16). IK is partly supported by the Hasso Plattner Institute Research School, the Israel Science Foundation (grant No. 1119/19), the Israeli Consortium for Network Programming (Neptune), the Technion Hiroshi Fujiwara Cyber Security Research Center and the Israel Cyber Bureau. GM is partly supported by the Hasso Plattner Institute Research School.

References

Andrews M, Kumaran K, Ramanan K, Stolyar A, Vijayakumar R, Whiting P (2004) Scheduling in a queuing system with asynchronously varying service rates. *Probability in the Engineering and Informational Sciences* 18(2):191–217.

Asmussen S (2008) *Applied probability and queues*, volume 51 (Springer Science & Business Media).

Daley D (1987) Certain optimality properties of the first-come first-served discipline for G/G/s queues. *Stochastic Processes and their Applications* 25:301–308.

Durrett R (2010) *Probability: theory and examples* (Cambridge university press).

Foschini G, Salz J (1978) A basic dynamic routing problem and diffusion. *IEEE Transactions on Communications* 26(3):320–327.

Foss S (1982) *Extremal problems in queueing theory.* Ph.D. thesis, PhD thesis, Novosibirsk State University, 1982. In Russian.

Foss S, Chernova N (1998) On the stability of a partially accessible multi-station queue with state-dependent routing. *Queueing Systems* 29(1):55–73.

Georgiadis L, Neely MJ, Tassiulas L, et al. (2006) Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking* 1(1):1–144.

Hajek B (2015) *Random processes for engineers* (Cambridge university press).

Koole GM (1992) *On the optimality of FCFS for networks of multi-server queues* (Centre for Mathematics and Computer Science).

Lu Y, Xie Q, Kliot G, Geller A, Larus JR, Greenberg A (2011) Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services. *Performance Evaluation* 68(11):1056–1071.

Malyshev VA, Men'shikov MV (1979) Ergodicity, continuity and analyticity of countable markov chains. *Trudy Moskovskogo Matematicheskogo Obshchestva* 39:3–48.

Meyn SP, Tweedie R (1994) State-dependent criteria for convergence of markov chains. *The Annals of Applied Probability* 149–168.

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

Shah D (2017) Private communication.

Shah D, Prabhakar B (2002) The use of memory in randomized load balancing. *IEEE ISIT*, 125.

Stolyar AL (2015) Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Systems* 80(4):341–361.

Tassiulas L, Ephremides A (1992) Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. on Automatic Control* 37(12):1936–1948.

Vvedenskaya ND, Dobrushin RL, Karpelevich FI (1996) Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problemy Peredachi Informatsii* 32(1):20–34.

Walton N (2013) Stability of maxweight-(α, g). *arXiv preprint arXiv:1301.3723* .

Weber RR (1978) On the optimal assignment of customers to parallel servers. *Journal of Applied Probability* 15(2):406–413.

Winston W (1977) Optimality of the shortest line discipline. *Journal of Applied Probability* 14(1):181–189.

Wolff RW (1987) Upper bounds on work in system for multichannel queues. *Journal of applied probability* 24(2):547–551.