

On the Persistent-Idle Load Distribution Policy Under Batch Arrivals and Random Service Capacity

Rami Atar⁽²⁾, Isaac Keslassy⁽²⁾, Gal Mendelson⁽¹⁾, Ariel Orda⁽²⁾, Shay Vargaftik⁽³⁾

1. Graduate School of Business, Stanford University
2. Viterbi Faculty of Electrical Engineering, Technion
3. VMware Research

Abstract

The *Persistent-Idle* (PI) load distribution policy was recently introduced as an appealing alternative to current low-communication load balancing techniques. In PI, servers only update the dispatcher when they become idle, and the dispatcher always sends jobs to the last server that reported being idle. PI is unique in that it does not seek to push the server queue lengths towards equalization greedily. Rather, it aggressively pulls the servers away from starvation. As a result, PI's analysis requires different tools than other load balancing approaches. So far, PI was proven to achieve the stability region for Bernoulli arrivals and deterministic and constant service capacities.

Our main contribution is proving that PI achieves the stability region in a model with batch arrivals and random service capacities. Proving this result requires developing tighter bounds on quantities of interest and proving the fulfillment of a generalized version of the state-dependent drift criteria previously considered. We also present PI-Split, which allows batches to be split among idle servers and prove its stability. Finally, we conduct simulations comparing PI and PI-Split to other load balancing policies. Simulation results indicate that both achieve excellent performance for the model under consideration for a wide range of parameters.

Keywords— Parallel Server Model; Load Balancing; Persistent-Idle; Stability; Lyapunov Drift

1 Introduction

The Persistent Idle (PI) load distribution policy was recently introduced in [1] as an appealing reduced-state load balancing scheme. It is designed for the parallel server model, in which jobs arrive at a dispatcher and are routed to one of n , possibly heterogeneous, servers. In PI, a job is routed to an idle server if there is one or to the last server a job was sent to, otherwise. Thus the dispatcher only uses idleness information to make its routing decisions making the communication cost at most a single message per job.

Despite this low communication overhead, PI was shown to achieve the stability region when servers are heterogeneous; namely, with possibly different processing rates. In their considered model the following assumption were made: at most one job may arrive at a time slot; each job's workload depends on the server it is assigned to and equals the number of time slots required to process it on that server; each server completes one unit of work at each time slot. Thus at most one departure can occur in each server at each time slot.

In this paper, we study PI in a considerably different, well-established model with batch arrivals and random service capacities (e.g. [2]). An additional variant of the policy is considered, called PI-Split, in which the dispatcher may split jobs belonging to the same batch to several servers. Our main results are that PI and PI-Split achieve the stability region in this model.

Proving that PI achieves the stability region in this model presents a significant challenge. In particular, the proof in [1] relies on sampling the Markov chain describing the state of the system at the random times at which new servers become idle, and proving a state-dependent drift criteria is fulfilled by a carefully chosen Lyapunov function. This proof heavily relies on the fact that, given the current state, the next sampling time is deterministic. This is no longer the case in the model we consider.

Our main contribution is proving the fulfillment of a generalized version of a state-dependent drift criteria (we discuss this at length in Section 3.2.1) and proving it is sufficient for stability. Our proof includes developing tighter bounds on quantities of interest and a method to relate the current queue length state to the time interval's duration until the next sampling time.

Finally, we present simulation results comparing PI and PI-Split to other load-distribution policies that were considered in [1]. Simulation results indicate that PI has a comparable performance to Join-the-Shortest-Queue (JSQ) up to high loads, and favorable performance compared to other well-known reduced-state load balancing schemes, i.e., Join-the-Idle-Queue (JIQ) [3], Power-of-choice (JSQ(2)) [4] and Power-of-Memory (JSQ(1,1)) [5].

The remainder of this paper is organized as follows. Section 2 contains the system model and the definition of the PI and PI-Split policies. Section 3 contains the stability results for PI and PI-Split. Section 4 contains the simulation results, comparing PI and PI-Split to other load-distribution policies.

2 Model

2.1 Persistent Idle

We consider a parallel-server system that evolves in discrete time $t \in \{0, 1, 2, \dots\}$ with a single dispatcher and n heterogeneous work-conserving servers. Each of the n servers has an unbounded FIFO queue holding pending jobs.

Order of events. The order of events during each time slot is as follows. (1) New jobs may arrive at the dispatcher; (2) the dispatcher immediately decides on a server to which the jobs will be assigned to and routes them accordingly; (3) servers provide service. In particular, servers can process jobs that have just arrived; (4) communication between the dispatcher and the servers may occur. As a result, the dispatcher may update its state (e.g. which servers have no pending jobs), on which it may base the assignment decision at the next time slot.

Batch arrivals. The job arrival process is given by a sequence of *i.i.d.* random variables (RVs) $\{a(t)\}_{t=1}^{\infty}$. Namely, at each time slot $t > 0$, $a(t)$ jobs arrive as a *batch* at the dispatcher. We assume $\mathbb{P}(a(1) = 0) > 0$ to avoid dealing with a periodic Markov chain. We denote the mean and standard deviation of $a(1)$ by λ and σ_a , respectively.

Random service capacities. The service process of server $i \in \{1, \dots, n\} := [n]$ is given by a sequence of *i.i.d.* RVs $\{s_i(t)\}_{t=1}^{\infty}$, representing the potential number of jobs that a server can complete in a time slot. We assume that these RVs are positive and bounded, i.e., there exists $s_{max} \in \mathbb{N}$ such that

$$1 \leq s_i(t) \leq s_{max}, \quad \text{for all } i \in [n] \text{ and } t > 0. \quad (1)$$

We denote the mean and standard deviation of $s_i(1)$ by μ_i and σ_{s_i} , respectively. Assumption (1) implies

$$1 \leq \mu_i \leq s_{max}, \quad \text{for all } i \in [n]. \quad (2)$$

Queue lengths and idleness. Let $Q_i(t)$ denote the number of pending jobs in queue i at the *end* of time slot t , *after* arrival and departure. Let $Q(t) = (Q_1(t), \dots, Q_n(t))$, and $Q = (Q(t))_{t=0}^{\infty}$. We refer to $Q(0)$ as the initial condition. Server i is *idle* at time slot t if $Q_i(t) = 0$. A server *becomes* idle at time slot $t > 0$ if it had jobs to process during time slot t and processed all of them. For example, if $Q_i(t-1) = 0$, jobs arrived to server i at the beginning of time slot t and it served all of them, i.e. $Q_i(t) = 0$, then it became idle at time slot t . But, if $Q_i(t-1) = 0$ and $Q_i(t) = 0$ and no jobs arrived at server i , it did not become idle at time slot t .

Tokens. A single token is associated with each server. Thus, the total number of tokens in the system is n . Each token can be held by either the corresponding server or the dispatcher. When a server becomes idle, it sends its token to the dispatcher. Similarly, when the dispatcher chooses an idle server as a destination for a job, it implicitly sends the token back to this server along with the job. At time slot $t = 0$ the dispatcher holds the tokens corresponding to the servers that are initially idle.

Example 1. If $Q_i(t-1) > 0$, then server i must hold its token at the beginning of time slot t . If, in addition, $Q_i(t) = 0$, server i becomes idle at time slot t and sends the token at the end of time slot t .

Example 2. If $Q_i(t-1) = 0$, the dispatcher must hold server i 's token at the beginning of time slot t . If, in addition, jobs arrive to server i at the beginning of time slot t then the dispatcher sends the token to server. If server i now completes processing all jobs, it sends the token at the end of time slot t . Thus, in this case, the token travels from the dispatcher and back at the same time slot.

PI routing policy. The routing is encoded in a process that takes values in $[n]$, denoted by $LI = (LI(t))_{t=0}^{\infty}$. If a job arrives at time $t > 0$, it joins queue $LI(t)$. We refer to the corresponding server as the *Last-Idle* server. Roughly speaking, the Last-Idle server is the one that was idle most recently. If there are multiple idle servers, the Last-Idle server is chosen by some tie-breaking rule. For concreteness we assume that the tie-breaking rule is that the Last-Idle server is chosen uniformly at random among the tied servers. The exact

mathematical definition of the process LI appears below. Thus, the policy persistently sends the arriving jobs to the Last-Idle server until other servers become idle.

Markov chain formulation. The precise mathematical formulation is given in terms of a Markov chain, that we now define. The Markov chain, denoted $\{X(t)\} = \{(Q(t), LI(t))\}$, has state space

$$\hat{\mathcal{A}} = \mathbb{N}^n \times [n].$$

A member α of the state space $\hat{\mathcal{A}}$ is of the form

$$\alpha = (\alpha_1, \dots, \alpha_n, l),$$

where $\alpha_1, \dots, \alpha_n$ are queue lengths, and l is the identity of the Last-Idle server.

Define the set of idle servers at the end of time slot t by

$$I(t) = \{i : Q_i(t) = 0\}. \quad (3)$$

The queue length and Last-Idle processes satisfy the following recursion for $t > 0$,

$$\begin{cases} Q_i(t) = [Q_i(t-1) + a(t)\mathbb{1}_{\{LI(t)=i\}} - s_i(t)]^+ \\ LI(t) = \begin{cases} LI(t-1), & \text{if } |I(t-1)| = 0 \\ \xi_t, & \text{otherwise,} \end{cases} \end{cases} \quad (4)$$

where for each $t > 0$, ξ_t is a RV whose conditional distribution given

$$(X(0), X(1), \dots, X(t-1), a(t), s_1(t), \dots, s_n(t))$$

is uniform on the set $I(t-1)$ on the event $\{|I(t-1)| > 0\}$, and $\xi_t = 1$ on the event $\{|I(t-1)| = 0\}$ (note that the value of ξ_t on the latter event is immaterial; it is specified only for concreteness). It is clear from the definition that X forms a Markov chain.

We define the set $\mathcal{A} \subset \hat{\mathcal{A}}$ as the collection of all states $\alpha \in \hat{\mathcal{A}}$ such that α can be reached with a positive probability from any of the states of the form $(0, \dots, 0, i)$, $i \in [n]$, corresponding to an empty system. The initial condition $X(0) = (Q(0), LI(0))$ is assumed to lie in \mathcal{A} with probability 1. Under this assumption, we may and will assume without loss of generality that the state space of X is \mathcal{A} .

By (4), the definition of \mathcal{A} and the fact that the process can reach $(0, \dots, 0, i)$ from $(0, \dots, 0, j)$ for any i, j (thanks to the assumptions that $\mathbb{P}(a(1) = 0) > 0$ and the uniformly-at-random tie-breaking rule), it follows that X is an irreducible, aperiodic, homogeneous Markov chain on \mathcal{A} .

Departures. Let $d_i(t)$ denote the number of jobs completed by server i during time slot t . Clearly, $d_i(t) \leq s_i(t)$. Moreover, $d_i(t)$ is given by

$$d_i(t) = \min\{s_i(t), Q_i(t-1) + a(t)\mathbb{1}_{\{LI(t)=i\}}\},$$

and the first equation in recursion (4) can also be written as

$$Q_i(t) = Q_i(t-1) + a(t)\mathbb{1}_{\{LI(t)=i\}} - d_i(t). \quad (5)$$

2.2 Persistent Idle with batch splitting

The PI policy sends all of the jobs belonging to a batch to a single server. The batch-splitting PI policy (referred to as *PI-Split*) is a variation in which whenever there are at least two idle servers, the dispatcher is allowed to split an arriving batch into *sub-batches* and send them to multiple idle servers. If there are none, the dispatcher sends the batch to the server corresponding to the value of $LI(t)$.

Under PI-Split, the tokens are handled exactly as under PI: when a server becomes idle, it sends its token to the dispatcher and when a dispatcher sends jobs to a server (either a batch or a sub-batch) it implicitly returns the token. Thus, under PI-Split, whenever a batch is split into sub-batches, multiple tokens are returned to the corresponding servers.

The updating rule for LI remains the same as under PI: If there are idle servers, LI is chosen uniformly at random. If there are none, its value remains unchanged. We add a superscript S for the batch splitting case such that the queue lengths, idleness and Last-Idle processes are given by Q^S , I^S and LI^S respectively. The state of the system is now given by the process

$$X^S =: (Q^S, LI^S).$$

The only difference between PI and PI-split is how the queue lengths are updated whenever $|I^S(t-1)| > 0$. To keep the Markov chain formulation simple for this case, we assume that the dispatcher determines how to split the batch between the idle servers based only on the current queue length state and the current batch size.

Formally, let $\{B(t)\}_{t=1}^\infty$ be an i.i.d process taking values in \mathbb{R} . Let $v[i]$ denote the i th member of a vector v in \mathbb{N}^n . Define a family of deterministic functions

$$f : \mathbb{N}^n \times \mathbb{N} \times \mathbb{R} \rightarrow \mathbb{N}^n \quad (6)$$

satisfying the following conditions:

$$\begin{aligned} \sum_{i=1}^n f(\alpha_1, \dots, \alpha_n, a, b)[i] &= a, \\ f(\alpha_1, \dots, \alpha_n, a, b)[i] &= 0 \quad \text{if } \min_{1 \leq j \leq n} \alpha_j = 0 \text{ and } \alpha_i > 0. \end{aligned} \quad (7)$$

The inputs to f are the queue lengths Q^S , the batch size $a(t)$ and the value of the i.i.d process $B(t)$. The i th member of the output vector equals the number of jobs the dispatcher should send to server i . The first condition in (7) states that the total number of assigned jobs is exactly the batch size. The second condition states that if there are idle servers ($\min_{1 \leq j \leq n} \alpha_j = 0$), then any server that is not idle should not receive any jobs. The behaviour of f in the case where there are no idle servers is immaterial.

Following the definitions above, we can now state the updating rule for X^S :

$$\begin{cases} Q_i^S(t) = [Q_i^S(t-1) + a_i(t) - s_i(t)]^+ \\ LI^S(t) = \begin{cases} LI^S(t-1), & \text{if } |I^S(t-1)| = 0 \\ \xi_t^S, & \text{otherwise,} \end{cases} \\ a_i(t) = \begin{cases} a(t), & \text{if } |I^S(t-1)| = 0 \text{ and } i = LI^S(t) \\ f(X_i^S(t-1), a(t), B(t))[i], & \text{if } |I^S(t-1)| > 0 \text{ and } i \in I^S(t-1) \\ 0, & \text{otherwise} \end{cases} \end{cases} \quad (8)$$

where ξ_t^S is a RV distributed uniformly on the idle servers and is defined analogously to ξ_t in (4) and the function f encodes the batch splitting rule and belongs to the family of functions defined by (6) and (7). Thus, X^S is an irreducible, aperiodic, time homogeneous Markov chain.

3 Main results

3.1 Communication overhead

We begin with a simple result regarding the communication overhead of PI and PI-Split. Both only rely on tokens passed from the servers to the dispatcher.

Proposition 3.1. *The communication overhead of PI (resp. PI-Split) consists of at most a single message per batch (resp. sub-batch).*

Proof. A necessary condition for a transmission of a token is the completion of a job. Since PI does not split batches, a completion of a job which results in a server becoming idle must be a completion of a batch. Similarly, for PI-Split, a message can be sent only when sub-batches are completed and not all completions result in a message. \square

3.2 Stability results

We introduce the following notation. For $\alpha \in \mathcal{A}$, we write $\mathbb{E}_\alpha[\cdot]$ and $\mathbb{P}_\alpha(\cdot)$ for the conditional expectation $\mathbb{E}[\cdot | X(0)=\alpha]$ and probability $\mathbb{P}(\cdot | X(0)=\alpha)$, respectively.

3.2.1 Overview

A useful approach to proving stability for irreducible Markov processes relies on the construction of a Lyapunov function \mathcal{L} , mapping the state space to $[0, \infty)$, that satisfies a drift condition within a single time step, of the form

$$\mathbb{E}_\alpha[\mathcal{L}(X(1))] - \mathcal{L}(\alpha) < -\varepsilon < 0, \quad (9)$$

such that the drift is negative and bounded away from zero at all states α outside of some finite set $A \subset \mathcal{A}$, and a finite expectation at all states $\alpha \in A$, namely $\mathbb{E}_\alpha[\mathcal{L}(X(1))] < \infty$. The existence of such a function ensures that provided the chain starts in the finite set A , the expected return time of the chain to A is finite. This, in turn, implies several stability properties, such as positive recurrence and convergence to a unique invariant measure (Lemma I.3.10 of [6]).

However, this approach does not appear to be directly applicable for the model under consideration. As discussed at length in [1], the main difficulty arises when considering states where one queue length is large while the others are close to zero. It is possible that the large queue length belongs to the Last-Idle server, which receives all of the work.

The proof in [1] relies on sampling the underlying Markov chain at (random) times at which the identity of the Last-Idle server may change, namely, the times at which a token is available at the dispatcher and it is not the last used token. If the service capacities are deterministic (as in [1]), the next such time is a function of the current state alone. Then, the calculation of the drift is performed with respect to time intervals that depend (deterministically) on the states the process traverses (as in [7], [8]).

At these sampling times the queue length 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{A}_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 (9), it is shown in [1] that

$$\mathbb{E}_\alpha[\mathcal{L}(X(\tau))] - \mathcal{L}(\alpha) < -\varepsilon\tau, \quad (10)$$

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

The random service capacities we consider in this paper further complicate the analysis. In this case, given the current state $X(0) = \alpha$, the next sampling time τ is random. Instead of (10), our method is based on showing that

$$\mathbb{E}_\alpha[\mathcal{L}(X(\tau))] - \mathcal{L}(\alpha) < -\varepsilon\mathbb{E}_\alpha[\tau], \quad (11)$$

for all but finitely many α in the set \mathcal{A}_b .

We analyze the stability of PI and PI-Split in Sections 3.3 and 3.4 respectively.

3.3 Stability of PI

Theorem 3.2. *Assume $\lambda < \sum_{i=1}^n \mu_i$. Then:*

- (i) X is positive recurrent. Consequently,
- (ii) X has a unique stationary distribution, denoted by π_X , and
- (iii) For any initial state $\alpha \in \mathcal{A}$ and any $B \subset \mathcal{A}$, $\mathbb{P}_\alpha(X(t) \in B) \rightarrow \pi_X(B)$ as $t \rightarrow \infty$.

Proof. For $A \subset \mathcal{A}$ define the hitting time

$$\tau^A = \inf \{t \geq 1; X(t) \in A\}. \quad (12)$$

We prove that there exists a nonempty finite set $A \subset \mathcal{A}$ such that for all $\alpha_0 \in A$,

$$\mathbb{E}_{\alpha_0}[\tau^A] < \infty. \quad (13)$$

By I.3.10 of [6], the irreducibility of X and (13) imply positive recurrence and hence part (i) of Theorem 3.2. Part (ii) then follows by I.3.6 of [6]. Finally, since the chain is aperiodic, part (iii) follows by the ergodic theorem for Markov chains, Theorem I.4.2 of [6]. It thus remains to show (13). To this end, we define a sampled chain Y of X , sampled at the times when the value of LI may change.

To define the random sampling times, we first define the following filtrations. The first is the natural filtration of X , namely

$$\mathcal{F}_t = \sigma(X(0), \dots, X(t)) \quad (14)$$

and the second also includes the RVs ξ_1, \dots, ξ_{t+1} which are used to break ties between the idle servers, namely

$$\mathcal{F}_t^\xi = \sigma(X(0), \dots, X(t), \xi_1, \dots, \xi_{t+1}). \quad (15)$$

The filtration \mathcal{F}_t^ξ in (15) is useful when knowing the identity of the newly used token simplifies the analysis.

By (4), the value of LI can only change at time t if there is an idle server which is not the Last-Idle server, corresponding to the event

$$\Omega_t := \{\exists i \text{ s.t. } Q_i(t) = 0, i \neq LI(t)\}. \quad (16)$$

Define the sequence of stopping times τ_0, τ_1, \dots by

$$\begin{aligned}\tau_0 &= \inf\{t \geq 1 : \mathbb{1}_{\Omega_t} = 1\} \\ \tau_{k+1} &= \inf\{t > \tau_k : \mathbb{1}_{\Omega_t} = 1\}, \quad k = 0, 1, 2, \dots\end{aligned}\tag{17}$$

Since the values of $\tau^A, \tau_0, \tau_1, \dots$ only depend on the values of X , the former are stopping times with respect to both filtrations, \mathcal{F}_t and \mathcal{F}_t^ξ . By the definition of the policy, the finiteness of the initial condition and the finiteness of the arrival process, the stopping times τ_k are finite a.s. for all $k \geq 0$ (the proof follows the same argument as in Lemma 2 of [1] and hence is omitted). Define the sampled chain

$$Y_k = X(\tau_k), \quad k \geq 0.\tag{18}$$

Denote by $\mathcal{A}^* \subset \mathcal{A}$ the collection of states that can be reached by Y , namely

$$\mathcal{A}^* = \{\alpha = (\alpha_1, \dots, \alpha_n, l) \in \mathcal{A} : \alpha_i = 0 \text{ for some } i \neq l\}.$$

The following lemma asserts that a certain drift criterion is fulfilled by Y .

Lemma 3.3. *There exist a constant γ such that $0 < \gamma \leq 1$, a function $\mathcal{L} : \mathcal{A} \rightarrow \mathbb{R}_+$ of the form*

$$\mathcal{L}(\alpha) = \sum_{i=1}^n \alpha_i^{1+\gamma},\tag{19}$$

a positive constant ϵ and a finite non-empty set $A \subset \mathcal{A}$, such that for all $k \geq 0$ and $\alpha \in \mathcal{A}^* \setminus A$, the following inequality holds:

$$\mathbb{E}_\alpha[(\mathcal{L}(Y_{k+1}) - \mathcal{L}(Y_k)) \mathbb{1}_{\{k < \sigma\}} \mid \mathcal{F}_{\tau_k}^\xi] \leq -\epsilon \mathbb{E}_\alpha[(\tau_{k+1} - \tau_k) \mathbb{1}_{\{k < \sigma\}} \mid \mathcal{F}_{\tau_k}^\xi],\tag{20}$$

where

$$\sigma = \inf\{k \geq 0 : Y_k \in A\}.\tag{21}$$

The stopping time σ counts the number of samples needed until Y hits the set A for the first time. Inequality (20) states that, given the initial condition is outside of A , as long as $k < \sigma$, the drift of the function \mathcal{L} applied to Y is strictly negative and is proportional to the expected duration of the intervals between samples. The proof is provided below the proof of the theorem.

Now, fix an initial state α_0 such that $X(0) = \alpha_0 \in A$. By the definition of the sampling times $\sigma, \tau_0, \tau_1, \dots$ in (17), we have

$$\tau_\sigma = \inf\{t \geq 1 : X(t) \in \mathcal{A}^* \cap A\},\tag{22}$$

such that $X(\tau_\sigma) = Y(\sigma) \in \mathcal{A}^* \cap A$. The value of τ_σ only depends on the values of X , hence τ_σ is a stopping time with respect to \mathcal{F}_t and \mathcal{F}_t^ξ . While X reaches A at time slot τ_σ , it is not necessarily the first time slot in $[1, \tau_\sigma]$ in which this occurs. Since τ^A , defined in (12), equals the number of time slots it takes X to reach the set A for the first time, we must have $\tau^A \leq \tau_\sigma$ and therefore

$$\mathbb{E}_{\alpha_0}[\tau^A] \leq \mathbb{E}_{\alpha_0}[\tau_\sigma].\tag{23}$$

Our goal is now to prove that $\mathbb{E}_{\alpha_0}[\tau_\sigma]$ is finite. We have:

$$\begin{aligned}\mathbb{E}_{\alpha_0}[\tau_\sigma] &= \sum_{\alpha \in \mathcal{A}^* \cap A} \mathbb{E}_{\alpha_0}[\tau_\sigma \mid X(\tau_0) = \alpha] \mathbb{P}_{\alpha_0}(X(\tau_0) = \alpha) \\ &\quad + \sum_{\alpha \in \mathcal{A}^* \setminus A} \mathbb{E}_{\alpha_0}[\tau_\sigma \mid X(\tau_0) = \alpha] \mathbb{P}_{\alpha_0}(X(\tau_0) = \alpha).\end{aligned}\tag{24}$$

Conditioned on the event $\{X(\tau_0) = \alpha \in \mathcal{A}^* \cap A\}$, by the definitions of τ_0 in (17) and τ_σ in (22), we have $\tau_\sigma = \tau_0$ and therefore

$$\mathbb{E}_{\alpha_0}[\tau_\sigma \mid X(\tau_0) = \alpha] = \mathbb{E}_{\alpha_0}[\tau_0 \mid X(\tau_0) = \alpha], \quad \alpha \in \mathcal{A}^* \cap A.\tag{25}$$

We proceed with analyzing the term $\mathbb{E}_{\alpha_0}[\tau_\sigma \mid X(\tau_0) = \alpha]$ for the case $\alpha \in \mathcal{A}^* \setminus A$. First, we prove that

$$\mathbb{E}_{\alpha_0}[\tau_\sigma \mid \mathcal{F}_{\tau_0}] = \tau_0 + g(X(\tau_0)),\tag{26}$$

where $g(x) := \mathbb{E}_x[\tau_\sigma]$.

Since τ_0 and τ_σ are stopping times with respect to \mathcal{F}_t and $\tau_0 \leq \tau_\sigma$ a.s., by the Strong Markov Property (Theorem 5.2.5. of [9]), we have

$$\mathbb{E}_{\alpha_0}[(\tau_\sigma - \tau_0) \wedge r \mid \mathcal{F}_{\tau_0}] = \phi^r(X(\tau_0)), \quad (27)$$

where r is a positive constant and $\phi^r(x) := \mathbb{E}_x[\tau_\sigma \wedge r]$. By the monotone convergence theorem we have

$$\mathbb{E}_x[\tau_\sigma \wedge r] \xrightarrow{r \rightarrow \infty} \mathbb{E}_x[\tau_\sigma] \quad (28)$$

and

$$\mathbb{E}_{\alpha_0}[(\tau_\sigma - \tau_0) \wedge r \mid \mathcal{F}_{\tau_0}] \xrightarrow{r \rightarrow \infty} \mathbb{E}_{\alpha_0}[(\tau_\sigma - \tau_0) \mid \mathcal{F}_{\tau_0}]. \quad (29)$$

Taking the limit w.r.t. r in both sides of (27) and using (28) and (29) yields (26).

Since $X(\tau_0)$ is measurable with respect to \mathcal{F}_{τ_0} , taking the conditional expectation of both sides of (26), conditioned on $X(\tau_0)$, yields

$$\mathbb{E}_{\alpha_0}[\mathbb{E}_{\alpha_0}[\tau_\sigma \mid \mathcal{F}_{\tau_0}] \mid X(\tau_0)] = \mathbb{E}_{\alpha_0}[\tau_\sigma \mid X(\tau_0)] = \mathbb{E}_{\alpha_0}[\tau_0 \mid X(\tau_0)] + g(X(\tau_0)),$$

which implies

$$\mathbb{E}_{\alpha_0}[\tau_\sigma \mid X(\tau_0) = \alpha] = \mathbb{E}_{\alpha_0}[\tau_0 \mid X(\tau_0) = \alpha] + \mathbb{E}_\alpha[\tau_\sigma]. \quad (30)$$

Substituting (25) and (30) in (24) yields

$$\begin{aligned} \mathbb{E}_{\alpha_0}[\tau_\sigma] &= \sum_{\alpha \in \mathcal{A}^*} \mathbb{E}_{\alpha_0}[\tau_0 \mid X(\tau_0) = \alpha] \mathbb{P}_{\alpha_0}(X(\tau_0) = \alpha) \\ &\quad + \sum_{\alpha \in \mathcal{A}^* \setminus A} \mathbb{E}_\alpha[\tau_\sigma] \mathbb{P}_{\alpha_0}(X(\tau_0) = \alpha) \\ &= \mathbb{E}_{\alpha_0}[\tau_0] + \sum_{\alpha \in \mathcal{A}^* \setminus A} \mathbb{E}_\alpha[\tau_\sigma] \mathbb{P}_{\alpha_0}(X(\tau_0) = \alpha), \end{aligned} \quad (31)$$

where the last equality is due to fact that the value of $X(\tau_0)$ is almost surely in \mathcal{A}^* . We proceed by showing that the two terms in the right hand side of (31) are finite, which, combined with (23) proves (13) and concludes the proof.

First, we consider the term $\mathbb{E}_{\alpha_0}[\tau_0]$. By (17), the stopping time τ_0 is the first time after time 0 such that one of the queue lengths, corresponding to a server different from $LI(0)$, equals zero. Under PI, the servers different from $LI(0)$ do not receive any work during $[1, \tau_0]$. By (1), a non-idle server must serve at least one job during a time slot. Thus,

$$\tau_0 \leq \max_i \{Q_i(0) : i \neq LI(0)\} \leq \max_i \{Q_i(0)\} \quad (32)$$

and therefore

$$\mathbb{E}_{\alpha_0}[\tau_0] \leq \max_i \{Q_i(0)\} < \infty, \quad (33)$$

where the last inequality is due to the finiteness of the initial condition.

Next, we consider the second term in the right hand side of (31). The proof proceeds by deriving an upper bound on $\mathbb{E}_\alpha[\tau_\sigma]$ for $\alpha \in \mathcal{A}^* \setminus A$ by using the sampled chain Y , defined in (18), and Lemma 3.3. Taking expectation on both sides of (20) yields

$$\mathbb{E}_\alpha[(\mathcal{L}(Y_{k+1}) - \mathcal{L}(Y_k)) \mathbb{1}_{\{k < \sigma\}}] \leq -\epsilon \mathbb{E}_\alpha[(\tau_{k+1} - \tau_k) \mathbb{1}_{\{k < \sigma\}}]. \quad (34)$$

We use (34) twice: to prove that τ_σ is well defined, namely, that $\sigma < \infty$ a.s., and then to derive a bound for $\mathbb{E}_\alpha[\tau_\sigma]$.

By the definition of the sampling times in (17), we have $\tau_{k+1} - \tau_k \geq 1$. Thus (34) implies that

$$\mathbb{E}_\alpha[(\mathcal{L}(Y_{k+1}) - \mathcal{L}(Y_k)) \mathbb{1}_{\{k < \sigma\}}] \leq -\epsilon \mathbb{E}_\alpha[\mathbb{1}_{\{k < \sigma\}}].$$

Summing over $k \in [0, m-1]$ for $m \geq 1$ yields

$$\mathbb{E}_\alpha[\mathcal{L}(Y_{m \wedge \sigma})] - \mathbb{E}_\alpha[\mathcal{L}(Y_0)] \leq -\epsilon \sum_{k=0}^{m-1} \mathbb{E}_\alpha[\mathbb{1}_{\{k < \sigma\}}] = -\epsilon \sum_{k=0}^{m-1} \mathbb{P}_\alpha(\sigma > k),$$

After rearranging we obtain

$$\sum_{k=0}^{m-1} \mathbb{P}_\alpha(\sigma > k) \leq \mathbb{E}_\alpha[\mathcal{L}(Y_0)]/\epsilon - \mathbb{E}_\alpha[\mathcal{L}(Y_{m \wedge \sigma})]/\epsilon \leq \mathbb{E}_\alpha[\mathcal{L}(Y_0)]/\epsilon = \mathbb{E}_\alpha[\mathcal{L}(X(\tau_0))]/\epsilon,$$

where in the second inequality we used the fact that $\mathbb{E}_\alpha[\mathcal{L}(Y_{m \wedge \sigma})] \geq 0$. Hence

$$\mathbb{E}_\alpha[\sigma] = \lim_{m \rightarrow \infty} \sum_{k=0}^{m-1} \mathbb{P}_\alpha(\sigma > k) \leq \mathbb{E}_\alpha[\mathcal{L}(X(\tau_0))]/\epsilon. \quad (35)$$

We now argue that $\mathbb{E}_\alpha[\mathcal{L}(X(\tau_0))]$ is finite for all initial states $\alpha \in \mathcal{A}$ (and therefore, specifically, for $\alpha \in \mathcal{A}^* \setminus \mathcal{A}$). Using the definition of \mathcal{L} in (19), we have

$$\mathbb{E}_\alpha[\mathcal{L}(X(\tau_0))] = \mathbb{E}_\alpha \left[\sum_{i=1}^n (Q_i(\tau_0))^{1+\gamma} \right] \leq \mathbb{E}_\alpha \left[\sum_{i=1}^n (Q_i(\tau_0))^2 \right], \quad (36)$$

where the last inequality is due to the fact that $0 < \gamma \leq 1$. Denote the maximum queue length at time t by $Q_M(t)$, namely, $Q_M(t) = \max_i \{Q_i(t)\}$. The value of $Q_M(0)$ is finite for all initial states $\alpha \in \mathcal{A}$. The quantities $Q_i(\tau_0)$ for servers different from the Last Idle server are each bounded from above by $Q_M(0)$. The queue length of the Last Idle server at τ_0 is bounded from above by $Q_M(0)$ plus the number of jobs that arrived to the system during $[1, \tau_0]$. Therefore, we have

$$\begin{aligned} \mathbb{E}_\alpha \left[\sum_{i=1}^n (Q_i(\tau_0))^2 \right] &\leq \mathbb{E}_\alpha \left[(n-1)Q_M^2(0) + \left(Q_M(0) + \sum_{t=1}^{\tau_0} a(t) \right)^2 \right] \\ &\leq \mathbb{E}_\alpha \left[(n-1)Q_M^2(0) + \left(Q_M(0) + \sum_{t=1}^{Q_M(0)} a(t) \right)^2 \right] \\ &\leq B \mathbb{E}_\alpha \left[Q_M^2(0) \right] = B \left(\max_{i \in [n]} \alpha_i \right)^2, \end{aligned} \quad (37)$$

for some constant $B > 0$, where the second inequality is due to (32) and the third inequality is due to the finite first and second moments of the arrival process and the fact that $Q_M(0)$ is an integer. Using (37) in (36) and recalling that Q_{max} is finite yields

$$\mathbb{E}_\alpha[\mathcal{L}(X(\tau_0))] \leq B \left(\max_{i \in [n]} \alpha_i \right)^2 < \infty, \quad \alpha \in \mathcal{A}. \quad (38)$$

Equations (35) and (38) establish that $\mathbb{E}_\alpha[\sigma] < \infty$ and consequently

$$\sigma < \infty \quad \text{a.s.} \quad (39)$$

Now, revisiting (34), summing over $k \in [0, m-1]$ yields

$$\mathbb{E}_\alpha[\mathcal{L}(Y_{m \wedge \sigma})] - \mathbb{E}_\alpha[\mathcal{L}(Y_0)] \leq -\epsilon \mathbb{E}_\alpha[\tau_{m \wedge \sigma}].$$

Using the fact that $\mathbb{E}_\alpha[\mathcal{L}(Y_{m \wedge \sigma})] \geq 0$ and rearranging yields

$$\mathbb{E}_\alpha[\tau_{m \wedge \sigma}] \leq \mathbb{E}_\alpha[\mathcal{L}(X(\tau_0))]/\epsilon. \quad (40)$$

Using (39), the monotone convergence theorem and (40), we obtain

$$\mathbb{E}_\alpha[\tau_\sigma] = \mathbb{E}_\alpha \left[\lim_{m \rightarrow \infty} \tau_{m \wedge \sigma} \right] = \lim_{m \rightarrow \infty} \mathbb{E}_\alpha[\tau_{m \wedge \sigma}] \leq \mathbb{E}_\alpha[\mathcal{L}(X(\tau_0))]/\epsilon \leq \bar{B} \left(\max_{i \in [n]} \alpha_i \right)^2, \quad (41)$$

where $\bar{B} > 0$ is some constant and in the last inequality we used (38). Using (23), (31) and (41) yields

$$\mathbb{E}_{\alpha_0}[\tau^A] \leq \mathbb{E}_{\alpha_0}[\tau_\sigma] \leq \mathbb{E}_{\alpha_0}[\tau_0] + \bar{B} \sum_{\alpha \in \mathcal{A}^* \setminus \mathcal{A}} \left(\max_{i \in [n]} \alpha_i \right)^2 \mathbb{P}_{\alpha_0}(X(\tau_0) = \alpha) \quad (42)$$

Recalling the definition of Q_M , the second term on the right hand side of (42) satisfies

$$\sum_{\alpha \in \mathcal{A}^* \setminus \mathcal{A}} \left(\max_{i \in [n]} \alpha_i \right)^2 \mathbb{P}_{\alpha_0}(X(\tau_0) = \alpha) \leq \sum_{\alpha \in \mathcal{A}^*} \left(\max_{i \in [n]} \alpha_i \right)^2 \mathbb{P}_{\alpha_0}(X(\tau_0) = \alpha) = \mathbb{E}_{\alpha_0} \left[Q_M^2(\tau_0) \right].$$

Therefore

$$\mathbb{E}_{\alpha_0}[\tau^A] \leq \mathbb{E}_{\alpha_0}[\tau_0] + \bar{B} \mathbb{E}_{\alpha_0} \left[Q_M^2(\tau_0) \right]. \quad (43)$$

Combining (33), (38) and (43) proves (13), which concludes the proof. \square

Proof of Lemma 3.3.

First, we discuss the term $\mathbb{1}_{\{k < \sigma\}}$ in (20). By (18), we have $(Y_0, \dots, Y_k) = (X(\tau_0), \dots, X(\tau_k))$. By (15), $\mathcal{F}_{\tau_k}^\xi$ contains the entire evolution of the process X during $[0, \tau_k]$ and therefore the values of the sampled chain Y, Y_0, \dots, Y_k , are measurable with respect to $\mathcal{F}_{\tau_k}^\xi$. Since σ (defined in (21)) corresponds to the first sample which lies in the finite set A , the values Y_0, \dots, Y_k are enough to determine whether or not the event $\{k < \sigma\}$ occurred. Thus, the random variable $\mathbb{1}_{\{k < \sigma\}}$ is also measurable with respect to $\mathcal{F}_{\tau_k}^\xi$, and we can write (20) as

$$\mathbb{1}_{\{k < \sigma\}} \mathbb{E}_\alpha[(\mathcal{L}(Y_{k+1}) - \mathcal{L}(Y_k)) \mid \mathcal{F}_{\tau_k}^\xi] \leq -\epsilon \mathbb{1}_{\{k < \sigma\}} \mathbb{E}_\alpha[(\tau_{k+1} - \tau_k) \mid \mathcal{F}_{\tau_k}^\xi], \quad (44)$$

Clearly, on the event $\{k \geq \sigma\}$, the two sides of inequality (44) equal zero and (20) holds. The rest of proof is dedicated to proving (20) on the event $\{k < \sigma\}$, namely, given that the values of Y_0, \dots, Y_k are all outside of A . For simplicity, we drop the subscript α from the expected value notation and write \mathbb{E} for \mathbb{E}_α for the rest of the proof.

We begin by fixing $\epsilon > 0$ and a function \mathcal{L} . We then consider a finite set A of the form

$$A = \{\alpha = (\alpha_1, \dots, \alpha_n, l) \in \mathcal{A} \mid \sum_i \alpha_i \leq \mathcal{C}\}, \quad (45)$$

and show that if \mathcal{C} is large enough, (20) holds.

Since $\lambda < \sum_{i=1}^n \mu_i$, there exists $\epsilon_0 > 0$ such that

$$\lambda - \sum_{i=1}^n \mu_i = -\epsilon_0.$$

Denote $\mu_{min} = \min_i \mu_i$ and let

$$\epsilon = (\epsilon_0 \wedge \mu_{min})/4. \quad (46)$$

Fix $0 < \gamma \leq 1$ such that

$$\max_{j \in \{1, \dots, n\}} \left\{ ([\lambda - \mu_j]^+)^{1+\gamma} - \sum_{i \neq j} \mu_i \right\} < -2\epsilon. \quad (47)$$

To prove the existence of such a γ , define $g_j(\gamma) = ([\lambda - \mu_j]^+)^{1+\gamma} - \sum_{i \neq j} \mu_i$. Now,

$$g_j(0) = [\lambda - \mu_j]^+ - \sum_{i \neq j} \mu_i = -(\epsilon_0 \wedge \sum_{i \neq j} \mu_i) \leq -(\epsilon_0 \wedge \mu_{min}) = -4\epsilon. \quad (48)$$

Define $h(\gamma) = \max_{j \in \{1, \dots, n\}} g_j(\gamma)$. By (48), $h(0) \leq -4\epsilon$. Since $g_j(\gamma)$ are continuous functions of γ , so is $h(\gamma)$. Thus, there exists $\gamma \in (0, 1]$ such that $h(\gamma) < -2\epsilon$, and (47) holds.

Recall (by (19)) the function $\mathcal{L} : \mathcal{A} \rightarrow \mathbb{R}_+$, for $\alpha = (\alpha_1, \dots, \alpha_n, l) \in \mathcal{A}$, is defined by

$$\mathcal{L}(\alpha) = \sum_{i=1}^n \alpha_i^{1+\gamma}. \quad (49)$$

Starting with the left hand side of (20), we use (49) to obtain

$$\mathbb{E}[\mathcal{L}(Y_{k+1}) - \mathcal{L}(Y_k) \mid \mathcal{F}_{\tau_k}^\xi] = \mathbb{E} \left[\sum_{i=1}^n \left(Q_i^{1+\gamma}(\tau_{k+1}) - Q_i^{1+\gamma}(\tau_k) \right) \mid \mathcal{F}_{\tau_k}^\xi \right]. \quad (50)$$

The proof proceeds by decomposing the sum in the right hand side of (50) into two parts and analyzing them separately: the first part corresponds to the Last Idle server at time $\tau_k + 1$, which we denote by

$$l_k = LI(\tau_k + 1). \quad (51)$$

The second part corresponds to the rest of the servers.

By (4), the value of l_k is determined using $X(\tau_k)$ and $\xi_{\tau_{k+1}}$, which, by (15), are included in $\mathcal{F}_{\tau_k}^\xi$. Thus, the random variable l_k is measurable with respect to $\mathcal{F}_{\tau_k}^\xi$. By the definition of the sampling times in (16) and (17), the identity of the Last Idle server remains unchanged during $[\tau_k + 1, \tau_{k+1}]$ and equals to l_k . Also,

$$Q_{l_k}(\tau_k) = 0. \quad (52)$$

Thus, during $[\tau_k + 1, \tau_{k+1}]$, the server corresponding to l_k receives all incoming jobs and its queue length is given by a reflected random walk starting at zero. The rest of the queue lengths decrease until the first reaches zero at time τ_{k+1} . Therefore, using the balance equation (4), we obtain

$$Q_i(\tau_{k+1}) = \left[Q_i(\tau_k) - \sum_{t=\tau_k+1}^{\tau_{k+1}} s_i(t) \right]^+, \quad i \in \{1, \dots, n\} \setminus \{l_k\}. \quad (53)$$

For simplicity, denote

$$\begin{aligned} Q_i^k &= Q_i(\tau_k), \quad i \in \{1, \dots, n\}, \\ R_i^k &= \sum_{t=\tau_k+1}^{\tau_{k+1}} s_i(t), \quad i \in \{1, \dots, n\} \setminus \{l_k\}. \end{aligned} \quad (54)$$

Using (51), (52), (53) and (54), we decompose the sum in the right hand side of (50) as follows:

$$\begin{aligned} \mathbb{E} \left[\sum_{i=1}^n \left(Q_i^{1+\gamma}(\tau_{k+1}) - Q_i^{1+\gamma}(\tau_k) \right) \mid \mathcal{F}_{\tau_k}^\xi \right] &= \mathbb{E} \left[\sum_{i \neq l_k}^n \left(([Q_i^k - R_i^k]^+)^{1+\gamma} - (Q_i^k)^{1+\gamma} \right) \mid \mathcal{F}_{\tau_k}^\xi \right] \\ &\quad + \mathbb{E} \left[\left(Q_{l_k}(\tau_{k+1}) \right)^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi \right]. \end{aligned} \quad (55)$$

Denote

$$\Delta_k = \tau_{k+1} - \tau_k. \quad (56)$$

We proceed by bounding the two terms in the right hand side of (55) separately. We begin with $\mathbb{E} \left[\left(Q_{l_k}(\tau_{k+1}) \right)^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi \right]$. Conditioned on $\mathcal{F}_{\tau_k}^\xi$, the duration of the interval $[\tau_k + 1, \tau_{k+1}]$, given by Δ_k , depends only on the queue lengths at the servers other than l_k and is therefore independent of the reflected random walk in server l_k . Denote

$$\begin{aligned} \beta_i &:= \lambda - \mu_i \\ \sigma_i^2 &:= \sigma_a^2 + \sigma_{s_i}^2. \end{aligned} \quad (57)$$

Using a modification of Lemma 6 of [1], we obtain

$$\mathbb{E} \left[\left(Q_{l_k}(\tau_{k+1}) \right)^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi \right] \leq ([\lambda - \mu_{l_k}]^+)^{1+\gamma} \mathbb{E}[\Delta_k^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi] + C \mathbb{E}[\Delta_k^{0.75(1+\gamma)} \mid \mathcal{F}_{\tau_k}^\xi], \quad (58)$$

where $C = \max_i \{ (16\sigma_i + 8(\sigma_i)^{1/2}(\beta_i)^+)^{(1+\gamma)/2} \}$. The proof is deferred to the Appendix.

Next, we consider the first term in the right hand side of (55). For ease of exposition, we suppress the index k and write Q_i and R_i for Q_i^k and R_i^k respectively. We have

$$\begin{aligned} ([Q_i - R_i]^+)^{1+\gamma} - Q_i^{1+\gamma} &= Q_i^{1+\gamma} \left(([1 - R_i/(Q_i \vee 1)]^+)^{1+\gamma} - 1 \right) \\ &\leq Q_i^{1+\gamma} \left([1 - R_i/(Q_i \vee 1)]^+ - 1 \right) \\ &= -Q_i^\gamma (Q_i \wedge R_i), \end{aligned} \quad (59)$$

where the inequality is due to the fact that $R_i/(Q_i \vee 1) \geq 0$ and thus $0 \leq [1 - R_i/(Q_i \vee 1)]^+ \leq 1$. Combining (50), (55), (58) and (59) we obtain

$$\mathbb{E}[\mathcal{L}(Y_{k+1}) - \mathcal{L}(Y_k) \mid \mathcal{F}_{\tau_k}^\xi] \leq -\epsilon \mathbb{E}[\Delta_k \mid \mathcal{F}_{\tau_k}^\xi] + \Phi_k, \quad (60)$$

where

$$\Phi_k := \mathbb{E} \left[\epsilon \Delta_k + ([\lambda - \mu_{l_k}]^+)^{1+\gamma} \Delta_k^{1+\gamma} + C \Delta_k^{0.75(1+\gamma)} - \sum_{i \neq l_k}^n Q_i^\gamma (Q_i \wedge R_i) \mid \mathcal{F}_{\tau_k}^\xi \right]. \quad (61)$$

Thus, using (60), (20) will follow once we prove that $\Phi_k \leq 0$ a.s. To this end, denote

$$\begin{aligned} Q_{min} &= \min_{1 \leq i \leq n, i \neq l_k} Q_i, \\ Q_{max} &= \max_{1 \leq i \leq n} Q_i. \end{aligned}$$

Denote by i_{max} the server with the maximum queue length at time τ_k , namely

$$i_{max} = \operatorname{argmax}_{1 \leq i \leq n} \{Q_i\}$$

Since the queue length at time τ_k of the Last Idle server l_k equals zero, we have $i_{max} \neq l_k$.

The key to analyzing (61) is to relate Δ_k , the duration of the interval $[\tau_k + 1, \tau_{k+1}]$, to the value of Q_{min} , which, by (15), is measurable with respect to $\mathcal{F}_{\tau_k}^\xi$. By the definition of the sampling times in (17) we have $\Delta_k \geq 1$. By (1), any strictly positive queue lengths of the servers different from l_k must decrease by at least one and by at most s_{max} at each time slot. Therefore, given $\mathcal{F}_{\tau_k}^\xi$, we have

$$(Q_{min}/s_{max}) \vee 1 \leq \Delta_k \leq Q_{min} \vee 1 \quad (62)$$

The argument proceeds by analyzing (61) in two cases, corresponding to the value of Q_{min} . The idea is that if Q_{min} is small, then by (62), so is Δ_k . Thus, if the finite set in (45) is chosen large enough so that Q_{max} must be very large, one of the members of $\sum_{i \neq l_k}^n Q_i^\gamma(Q_i \wedge R_i)$ is very large compared to the other terms in the right hand side of (61), resulting in $\Phi_k \leq 0$. Otherwise, if Q_{min} is large, then by (62) so is Δ_k . In this case, we prove that

$$\mathbb{E} \left[\sum_{i \neq l_k}^n Q_i^\gamma(Q_i \wedge S_i) \mid \mathcal{F}_{\tau_k}^\xi \right] \geq \mathbb{E} \left[\left(\sum_{i \neq l_k}^n \mu_i \right) \Delta_k^{1+\gamma} - (n-1)s_{max} \Delta_k^\gamma \mid \mathcal{F}_{\tau_k}^\xi \right].$$

Thus, if we use this to further bound Φ_k from above, by (47), the coefficient of the leading term $\Delta_k^{1+\gamma}$ is negative. If Δ_k is large enough we again obtain $\Phi_k \leq 0$.

Define

$$f(x) = \epsilon x - 2\epsilon x^{1+\gamma} + Cx^{0.75(1+\gamma)} + (n-1)s_{max}x^\gamma, \quad (63)$$

where C is from (58). Since the coefficient of the leading term $x^{1+\gamma}$ in $f(x)$ is negative, there exists $u > 0$ such that

$$u > s_{max}^2 \quad \text{and} \quad f(x) < 0 \quad \text{if } x > u/s_{max}. \quad (64)$$

Fix such u .

Choose the constant \mathcal{C} in the definition of the finite set A in (45) such that

$$\mathcal{C} > ns_{max}u + n \left(\epsilon u + (\max_i \{[\lambda - \mu_i]^+\})^{1+\gamma} u^{1+\gamma} + Cu^{0.75(1+\gamma)} \right)^{1/\gamma} + n \left(\epsilon + n(\sigma_a^2 + \lambda^2) \right)^{1/\gamma}. \quad (65)$$

By (45), for any state outside of A we have $\mathcal{C} < \sum_i Q_i \leq nQ_{max}$. Thus, by (65), we obtain

$$Q_{max} > s_{max}u + \left(\epsilon u + (\max_i \{[\lambda - \mu_i]^+\})^{1+\gamma} u^{1+\gamma} + Cu^{0.75(1+\gamma)} \right)^{1/\gamma}. \quad (66)$$

We now consider two cases that correspond to the value of Q_{min} , which, by (62), is related to the duration of the interval. The first two members of the right hand side of (65) correspond to our requirements of the finite set A for dealing with the aforementioned two cases. The third member is used for the proof of Theorem (3.4) on PI-Split.

Case 1: (short interval) $Q_{min} \leq u$.

In this case, by using (62), we obtain

$$\Delta_k \leq u \quad \text{if } Q_{min} \leq u, \quad (67)$$

namely, the duration of the interval is at most u . Since by (66) $Q_{max} > s_{max}u$ and s_{max} is the maximum number of jobs that can be served during a single time slot, the largest queue length at τ_k (in server i_{max}) does not reach zero at τ_{k+1} . By (1) the queue length in server i_{max} must decrease by at least one. Thus, using (54), we conclude that $R_{i_{max}} \geq 1$ and $Q_{max} \wedge R_{i_{max}} \geq 1$. Therefore, we have

$$\sum_{i \neq l_k}^n Q_i^\gamma(Q_i \wedge R_i) \geq Q_{max}^\gamma(Q_{max} \wedge R_{i_{max}}) \geq Q_{max}^\gamma. \quad (68)$$

Using (67) and (68) to upper bound the right hand side of (61) yields

$$\Phi_k \leq \mathbb{E} \left[\epsilon u + ([\lambda - \mu_{l_k}]^+)^{1+\gamma} u^{1+\gamma} + Cu^{0.75(1+\gamma)} - Q_{max}^\gamma \mid \mathcal{F}_{\tau_k}^\xi \right] < 0,$$

where the last inequality is due to (66).

Case 2: (long interval) $Q_{min} > u$.

By the definition of the sampling times, R_i can be larger than Q_i by at most $s_i(\tau_{k+1})$ (the number of jobs potentially served by server i at the last time slot of the interval $[\tau_k + 1, \tau_{k+1})$), which we assumed in (1) is bounded from above by s_{max} . Thus

$$Q_i \wedge R_i \geq (R_i - s_{max}) \wedge R_i = R_i - s_{max}.$$

Therefore

$$\mathbb{E} \left[\sum_{i \neq l_k}^n Q_i^\gamma(Q_i \wedge R_i) \mid \mathcal{F}_{\tau_k}^\xi \right] \geq \mathbb{E} \left[\sum_{i \neq l_k}^n Q_i^\gamma(R_i - s_{max}) \mid \mathcal{F}_{\tau_k}^\xi \right] = \sum_{i \neq l_k}^n Q_i^\gamma(\mathbb{E}[R_i \mid \mathcal{F}_{\tau_k}^\xi] - s_{max}), \quad (69)$$

where the last equality is due to the fact that Q_i is measurable with respect to $\mathcal{F}_{\tau_k}^\xi$ for all i .

Next, we calculate $\mathbb{E}[R_i \mid \mathcal{F}_{\tau_k}^\xi]$. By the definitions of R_i in (54) and Δ_k in (56), this is the expected value of a random walk with strictly positive step sizes, distributed as $s_i(0)$, starting at time $\tau_k + 1$ and stopped after Δ_k time slots. If Δ_k only depended on the queue length at server i , then it would have been a stopping time, and by Wald's Equation and the finiteness of the sampling times the expected value would have been equal to $\mu_i \mathbb{E}[\Delta_k \mid \mathcal{F}_{\tau_k}^\xi]$. However, this is not the case here, since Δ_k depends on all servers different from l_k . Even so, we are able to prove the same result holds. Using the definition of R_i in (54) and the fact that $\tau_{k+1} = \tau_k + \Delta_k$ we obtain

$$\begin{aligned} \mathbb{E}[R_i \mid \mathcal{F}_{\tau_k}^\xi] &= \mathbb{E} \left[\sum_{t=\tau_k+1}^{\tau_{k+1}} s_i(t) \mid \mathcal{F}_{\tau_k}^\xi \right] = \mathbb{E} \left[\sum_{t=\tau_k+1}^{\tau_k+\Delta_k} s_i(t) \mid \mathcal{F}_{\tau_k}^\xi \right] \\ &= \mathbb{E} \left[\sum_{t=\tau_k+1}^{\tau_k+Q_{min}} s_i(t) \mathbb{1}_{\{t \leq \tau_k+\Delta_k\}} \mid \mathcal{F}_{\tau_k}^\xi \right] \\ &= \mathbb{E} \left[\sum_{m=1}^{Q_{min}} s_i(\tau_k+m) \mathbb{1}_{\{m \leq \Delta_k\}} \mid \mathcal{F}_{\tau_k}^\xi \right], \end{aligned} \quad (70)$$

where the second to last equality is due to the fact that by (62), we have $\Delta_k \leq Q_{min}$, and the last equality is a simple variable change.

Since the random variables Q_{min} and τ_k are both measurable with respect to $\mathcal{F}_{\tau_k}^\xi$ and are finite a.s., we can change the order of the summation and the expectation, yielding

$$\mathbb{E} \left[\sum_{m=1}^{Q_{min}} s_i(\tau_k+m) \mathbb{1}_{\{m \leq \Delta_k\}} \mid \mathcal{F}_{\tau_k}^\xi \right] = \sum_{m=1}^{Q_{min}} \mathbb{E}[s_i(\tau_k+m) \mathbb{1}_{\{m \leq \Delta_k\}} \mid \mathcal{F}_{\tau_k}^\xi]. \quad (71)$$

The event $\{m \leq \Delta_k\}$ is equivalent to the event $\{\Delta_k \leq m-1\}^c$. Thus, given $\mathcal{F}_{\tau_k}^\xi$, whether or not the event $\{m \leq \Delta_k\}$ occurred is determined by the values of all the service processes, in all servers, during the interval $[\tau_k + 1, \dots, \tau_k + m - 1]$, which are independent of $s_i(\tau_k + m)$. Therefore

$$\begin{aligned} \sum_{m=1}^{Q_{min}} \mathbb{E}[s_i(\tau_k+m) \mathbb{1}_{\{m \leq \Delta_k\}} \mid \mathcal{F}_{\tau_k}^\xi] &= \sum_{m=1}^{Q_{min}} \mathbb{E}[s_i(\tau_k+m) \mid \mathcal{F}_{\tau_k}^\xi] \mathbb{E}[\mathbb{1}_{\{m \leq \Delta_k\}} \mid \mathcal{F}_{\tau_k}^\xi] \\ &= \sum_{m=1}^{Q_{min}} \mu_i \mathbb{E}[\mathbb{1}_{\{m \leq \Delta_k\}} \mid \mathcal{F}_{\tau_k}^\xi] = \mu_i \mathbb{E}[\Delta_k \mid \mathcal{F}_{\tau_k}^\xi], \end{aligned} \quad (72)$$

where the second equality is due to $s_i(\tau_k + m)$ being independent of $\mathcal{F}_{\tau_k}^\xi$ and the last equality is due to the fact that $\sum_{m=1}^{Q_{min}} \mathbb{1}_{\{m \leq \Delta_k\}} = \Delta_k$.

Combining (69), (70), (71) and (72) yields

$$\mathbb{E} \left[\sum_{i \neq l_k}^n Q_i^\gamma(Q_i \wedge R_i) \mid \mathcal{F}_{\tau_k}^\xi \right] \geq \sum_{i \neq l_k}^n Q_i^\gamma(\mu_i \mathbb{E}[\Delta_k \mid \mathcal{F}_{\tau_k}^\xi] - s_{max}) = \sum_{i \neq l_k}^n \mathbb{E}[Q_i^\gamma(\mu_i \Delta_k - s_{max}) \mid \mathcal{F}_{\tau_k}^\xi]. \quad (73)$$

Since we consider the case where $Q_{min} > u$, and by (64) $u > s_{max}^2$, by (62) we obtain

$$\Delta_k > Q_{min}/s_{max} > u/s_{max} > s_{max}. \quad (74)$$

Thus, using the fact that by (2), $\mu_i \geq 1$ for all i , we obtain

$$\mu_i \Delta_k - s_{max} > 0. \quad (75)$$

Using (73), (75) and the fact that $\Delta_k \leq Q_{min} \leq Q_i$ for $i \neq l_k$, we obtain

$$\begin{aligned} \mathbb{E} \left[\sum_{i \neq l_k}^n Q_i^\gamma (Q_i \wedge R_i) \mid \mathcal{F}_{\tau_k}^\xi \right] &\geq \sum_{i \neq l_k}^n \mathbb{E} [\Delta_k^\gamma (\mu_i \Delta_k - s_{max}) \mid \mathcal{F}_{\tau_k}^\xi] = \sum_{i \neq l_k}^n \mathbb{E} [\mu_i \Delta_k^{1+\gamma} - s_{max} \Delta_k^\gamma \mid \mathcal{F}_{\tau_k}^\xi] \\ &= \mathbb{E} \left[\left(\sum_{i \neq l_k}^n \mu_i \right) \Delta_k^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi \right] - (n-1) s_{max} \mathbb{E} [\Delta_k^\gamma \mid \mathcal{F}_{\tau_k}^\xi]. \end{aligned} \quad (76)$$

Using (76) to bound the right hand side of (61) from above, we obtain

$$\begin{aligned} \Phi_k &\leq \mathbb{E} \left[\epsilon \Delta_k + \left(([\lambda - \mu_{l_k}]^+)^{1+\gamma} - \sum_{i \neq l_k}^n \mu_i \right) \Delta_k^{1+\gamma} + C \Delta_k^{0.75(1+\gamma)} + (n-1) s_{max} \Delta_k^\gamma \mid \mathcal{F}_{\tau_k}^\xi \right] \\ &\leq \mathbb{E} \left[\epsilon \Delta_k - 2\epsilon \Delta_k^{1+\gamma} + C \Delta_k^{0.75(1+\gamma)} + (n-1) s_{max} \Delta_k^\gamma \mid \mathcal{F}_{\tau_k}^\xi \right] = \mathbb{E} [f(\Delta_k)] \mid \mathcal{F}_{\tau_k}^\xi, \end{aligned}$$

where we have used the definitions of γ in (47) and $f(\cdot)$ in (63). Finally, by (74) $\Delta_k > u/s_{max}$, so by (64) we have $f(\Delta_k) < 0$ yielding $\Phi_k \leq 0$. This concludes the proof. \square

3.4 Stability of PI-Split

Theorem 3.4. Assume $\lambda < \sum_{i=1}^n \mu_i$. Then:

- (i) X^S is positive recurrent. Consequently,
- (ii) X^S has a unique stationary distribution, denoted by π_X^S , and
- (iii) For any initial state $\alpha \in \mathcal{A}$ and any $B \subset \mathcal{A}$, $\mathbb{P}_\alpha(X^S(t) \in B) \rightarrow \pi_X^S(B)$ as $t \rightarrow \infty$.

Proof. The proof is similar to that of Theorem (3.2). Specifically, we prove that the state dependent drift in Lemma (3.3) is also fulfilled by the sampled chain Y^S , with the same finite set A (45), constants ϵ (46), γ (47) and the Lyapunov function \mathcal{L} (49), but with stopping times and a filtration which correspond to the chain X^S . Theorem (3.4) then follows from the same arguments made in the proof of Theorem (3.2).

Recall that \mathcal{A}^* denotes the collection of states in which there is at least one queue length equal to zero that does not correspond to the Last Idle server. For initial states $\alpha \in \mathcal{A}^* \setminus A$ in which *there is only one such queue length*, the behavior of PI and PI-Split is identical since batches are not split until (at least) the next sampling time. Thus, the analysis in the proof of Lemma (3.3) proves that the desired state dependent drift holds for these initial states.

It remains to consider initial states $\alpha \in \mathcal{A}^* \setminus A$ such that there are at least two queue lengths equal to zero which do not correspond to the Last-Idle server. By the definition of the sampling times, the next sampling time must be after exactly 1 time slot. Thus, it remains to prove that

$$\mathbb{E}_\alpha \left[\mathcal{L}(X^S(1)) - \mathcal{L}(X^S(0)) \right] = \sum_{i=1}^n \mathbb{E}_\alpha \left[(Q_i^S(1))^{1+\gamma} - (Q_i^S(0))^{1+\gamma} \right] \leq -\epsilon. \quad (77)$$

Decompose the sum after the first equality into three parts:

Part 1: the single member of the sum corresponding to the server with the maximum queue length at time slot 0 (if there are several, choose one arbitrarily). Denote the server with the maximal queue length by $j^* = \operatorname{argmax}_{1 \leq i \leq n} Q_i^S(0)$. By the definition of PI-Split, server j^* does not receive jobs at the beginning of time slot 1 because its queue length is positive and there are other servers which are idle. By the definition of the finite set in (45) and (65), $Q_{j^*}^S(0) \geq s_{max}$. Hence, the queue length at server j^* cannot reach zero after one time slot and we have $Q_{j^*}^S(1) = Q_{j^*}^S(0) - s_{j^*}(1)$.

Part 2: the members of the sum corresponding to servers with queue length equal to zero at time slot 0. The queue lengths of these servers at time slot 1 is upper bounded by the number of jobs that arrive at the beginning of time slot 1, namely, $a(1)$. Thus, since $\gamma \leq 1$, $a^2(1)$ is an upper bound for the contribution of each member of part 2 to the sum in (77).

Part three: the members of the sum not included in parts 1 and 2. By the definition of PI-Split, members of part 3 do not receive any work. Thus, their contribution to the sum in (77) is non positive. Specifically, it is

upper bounded by $a^2(1)$.

We conclude that

$$\begin{aligned}
\mathbb{E}_\alpha \left[\mathcal{L}(X^S(1)) - \mathcal{L}(X^S(0)) \right] &\leq \mathbb{E}_\alpha \left[(Q_{j^*}^S(1))^{1+\gamma} - (Q_{j^*}^S(0))^{1+\gamma} \right] + (n-1)\mathbb{E}_\alpha[a^2(1)] \\
&= \mathbb{E}_\alpha \left[(Q_{j^*}^S(0) - s_{j^*}(1))^{1+\gamma} - (Q_{j^*}^S(0))^{1+\gamma} \right] + n(\sigma_a^2 + \lambda^2) \\
&\leq (Q_{j^*}^S(0) - 1)^{1+\gamma} - (Q_{j^*}^S(0))^{1+\gamma} + n(\sigma_a^2 + \lambda^2) \\
&\leq -(Q_{j^*}^S(0))^\gamma + n(\sigma_a^2 + \lambda^2), \tag{78}
\end{aligned}$$

where in the second inequality we used the fact that by (1), $s_{j^*}(1) \geq 1$ and that given the initial state α , j^* and $Q_{j^*}^S(0)$ are known. The last inequality follows from the same calculation as is done in (59). Finally, by the choice of the finite set in (65) (specifically, the third member in the right hand side of (65)), we obtain

$$\mathbb{E}_\alpha \left[\mathcal{L}(X^S(1)) - \mathcal{L}(X^S(0)) \right] \leq -\epsilon,$$

which concludes the proof. \square

4 Simulations

The main purpose of this section is to complement the simulation results presented in [1] for a model with batch arrivals and random service capacities. To this end, we compare the performance of PI and PI-Split with that of the token-based JIQ as well as several other load-distribution policies, namely JSQ, JSQ(2) and JSQ(1,1).

We consider systems with 100 heterogeneous servers and study two cases corresponding to whether arriving batches can be split or not. When batches cannot be split, the aforementioned policies route the entire batch to a single queue. When batches can be split, it is not straight-forward how the policies should split them. There are many sensible ways in which this can be done. In our simulations we chose the following splitting schemes, based on the ‘‘water-filling’’ approach (see, for example, [2]):

- **JSQ:** Uses the water-filling approach where the jobs belonging to the batch are sent to the servers with the smallest queue lengths in an attempt to equalize them as much as possible. For example, if there are 3 servers with queue lengths (1, 2, 4) and a batch arrives with 3 jobs, server 1 will get 2 jobs and server 2 will get 1 job, resulting in the queue lengths (3, 3, 4). Ties are broken randomly.
- **JSQ(2):** Uses the same water-filling approach as JSQ, but only for the two servers that were chosen uniformly at random.
- **JSQ(1,1):** Same as JSQ(2), but for the ‘‘remembered’’ server and the one chosen uniformly at random.
- **JIQ:** Uses water-filling only for the idle servers. Specifically, a batch is split as evenly as possible between the idle servers. If there are no idle servers, each job in the batch is sent to a randomly chosen server.
- **PI:** If there are idle servers, the splitting is the same as in JIQ. Otherwise, the entire batch is sent to the last idle server.

As shall be seen throughout this section, simulation results indicate that PI is indeed stable and offers appealing properties in comparison to JIQ and other reduced-state policies.

4.1 Simulation settings

We consider a system with 100 heterogeneous servers, some being slow servers, working at a rate $\mu^{-1} = 1$, and the others being fast servers, working at a rate $(10\mu)^{-1} = 1/10$. The service time distribution of all servers is uniform and lower bounded by 1. Note that these service time distributions concur with Assumption (1). We implement three scenarios, which correspond to different partitions of the servers into the two classes. More specifically, we consider the following slow:fast ratios: 10:90, 50:50 and 90:10. The number of arrivals at each time slot is drawn from a Poisson process of rate λ . We are interested in values of λ such that $0 < \lambda < \sum_{i=1}^n \mu_i$, *i.e.*, any strictly positive and sub-critical arrival rates.

For each scenario, we run simulations for 10^6 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

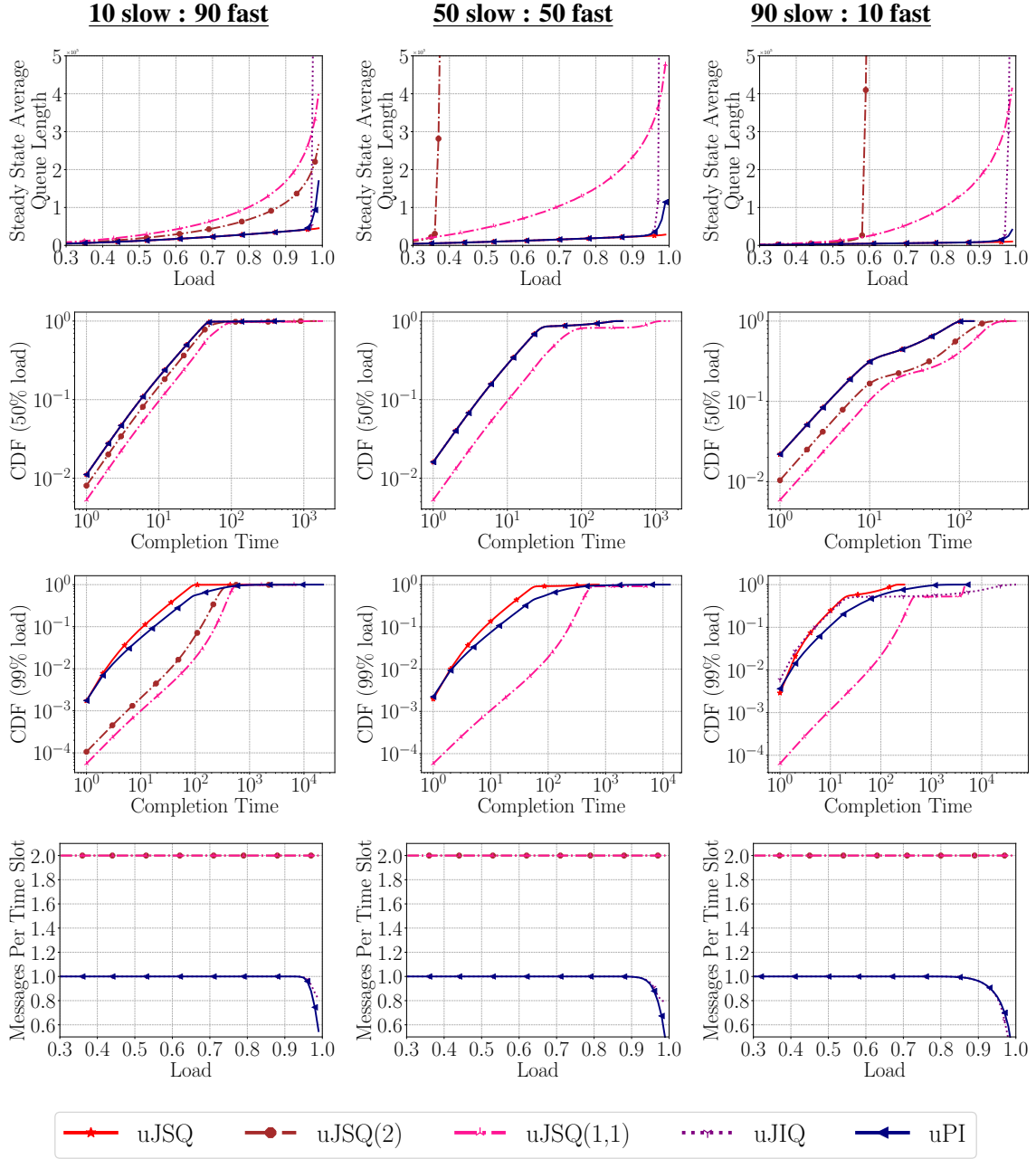


Figure 1: Performance of different policies with 100 heterogeneous servers and different slow:fast server ratios. The policies we consider here cannot split arriving batches of jobs. We emphasize this by adding 'u' before each policy name (which stands for 'unsplittable').

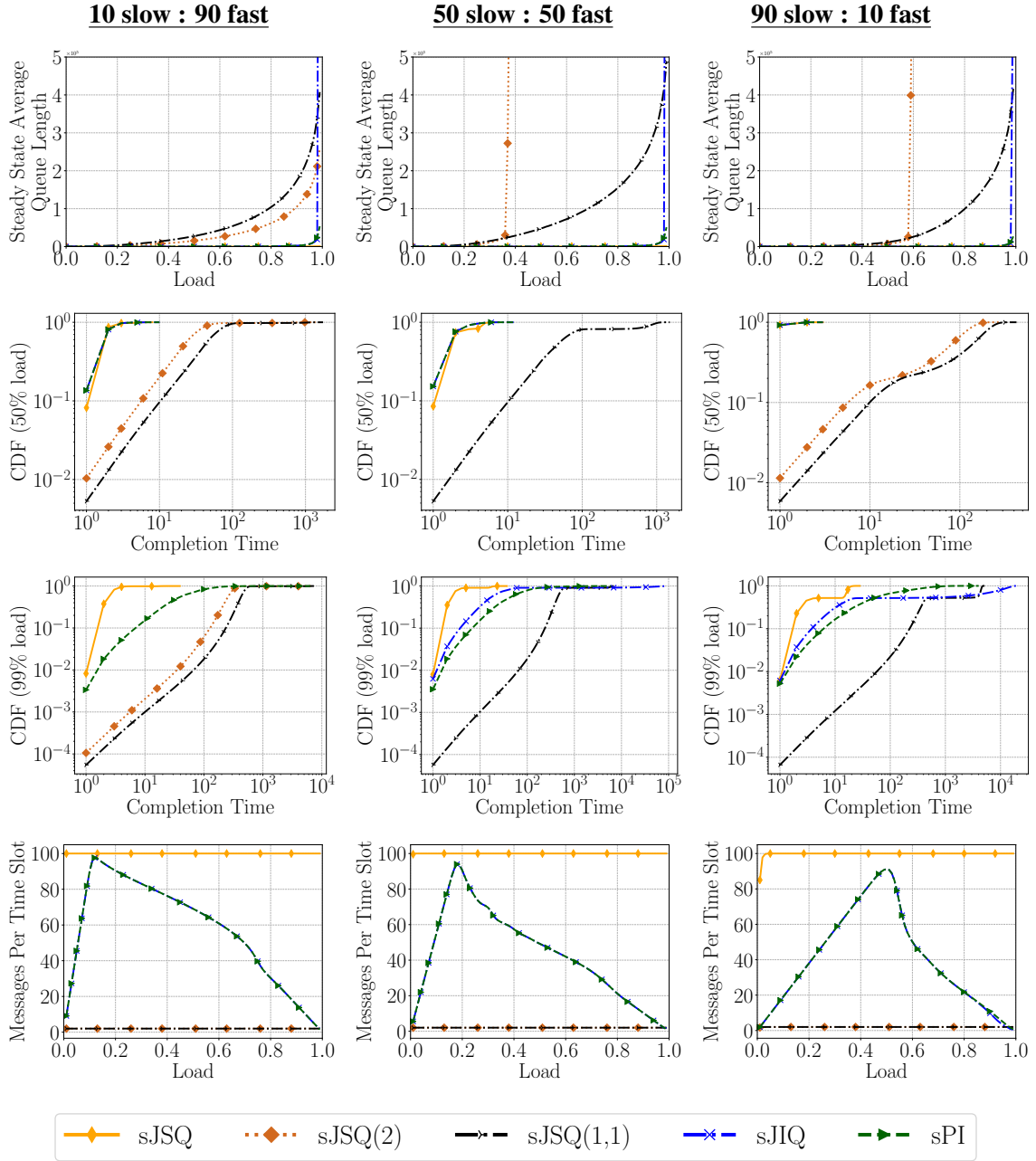


Figure 2: Performance of different policies with 100 heterogeneous servers and different slow:fast server ratios. The policies we consider here can split arriving batches of jobs. We emphasize this by adding 's' before each policy name (which stands for 'splittable').

such that the difference in the outputs of different runs at the maximal load was negligible. We present the following graphs:

Steady state average queue length vs. load. For each simulation run with a given load, we calculate the time averaged sum of queue lengths in the system. Under all policies, whenever the chain is positive recurrent, it is also ergodic. Thus the average queue length converges, and one simulation run is enough to calculate the steady-state average queue length. 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 queue lengths, these missing points indicate either poor performance or instability.

CDF of JCT in steady state for 50% and 99% load. We keep track of the job completion times of all jobs that entered the system during the simulations, with $\lambda=0.5 \sum_{i=1}^n \mu_i$ and $\lambda=0.99 \sum_{i=1}^n \mu_i$. We calculate the cumulative distribution function (CDF). Policies that are unstable under a given load are not shown in the graph.

Messages per time slot 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 this by the number of time slots to obtain the average number of messages per time slot. Counting messages for the other policies is more straightforward. At each time slot with non-zero arrivals, for JSQ it is the number of servers needed to be probed, *i.e.*, 100. Similarly, for JSQ(2) and JSQ(1,1) it is 2.

Remark 4.1. *Instead of calculating messages per time slot, one can choose to look at messages per batch, per sub batches (in the splittable case) or per job. However, measuring the number of messages per time slot is more informative for the amount of needed communication, whereas the other options are affected by the arrival and departure rates rather than by their ratio (i.e., normalized load).*

4.2 The unsplittable case: simulation results

Stability and average queue lengths. The results are depicted in Figure 1. As can be seen, for all the different slow:fast ratios, PI appears to achieve the stability region. Also, the average queue length under PI is favorable to that of other reduced-state policies and even comparable to that of JSQ for low and moderate loads. JSQ(1,1) also appears to achieve the stability region, but with a much larger average queue length. JSQ(2) appears to be unstable for moderately-high loads for the 50:50 and 90:10 ratios. Also, whenever JSQ(2) is stable, its average queue length is much higher than under PI. JIQ appears unstable for loads close to 100% for all ratios.

Job completion time. We measure the completion time of all jobs during the simulation at a moderate load of 50% and a high load of 99%. It is evident how both PI and JIQ are competitive to JSQ at 50% for all slow:fast ratios. This is because at such load there are many idle servers and both of these token-based policies resemble JSQ. JSQ(2) and JSQ(1,1) show lesser performance, and at a 50:50 ratio JSQ(2) is missing as its completion times diverge due to apparent instability. At a load of 99%, PI shows the best overall performance after JSQ. Both JIQ and JSQ(2) do not appear in 2 out of 3 scenarios due to what appears to be instability at this load. JSQ(1,1) again shows lesser performance.

Messages per time slot. Since the probability of zero arrivals in a time slot is negligible even at low loads (*i.e.*, $e^{-\lambda}$), the message overhead of JSQ, JSQ(2) and JSQ(1,1) is approximately constant. It is 100 for JSQ and 2 for JSQ(2) and JSQ(1,1). On the other hand, the number of messages per time slot for PI and JIQ is 1 up to a high load. As the load increases, there is an increasing number of time slots without any servers becoming idle and the message overhead for both policies decreases.

4.3 Splittable case: simulation results

Stability and average queue length. The results are depicted in Figure 2. As can be seen, the results show a similar trend to those of the unsplittable case. It is also evident how the advantage of JSQ in comparison to the other policies increases.

Job completion time and Messages per time slot. In the splittable case, the communication overhead of token based policies has different characteristics than in the unsplittable case. This is because in the splittable case, the overhead may reach 1 message per job rather than 1 message per batch of jobs. Indeed, this would be the case for moderate loads at which we have both a sufficient number of jobs in each batch and they are split among several/many idle servers.

As can be seen from the graphs, at such moderate loads, both PI and JIQ are competitive with JSQ but with often lower message overhead. JSQ(2) and JSQ(1,1) perform significantly worse but also have a lower message overhead in this case. At the high load of 99% the message overhead of the token based policies

rapidly decreases due to the decreased number of idled servers. In this case the results are similar to those of the unspittable case with apparent instability of JSQ(2) in 2 out of 3 cases and JIQ in 1 out of 3. JSQ(1,1) appears to be stable but again performs poorly.

Remark 4.2. *We have found in our simulations that the stability region of JIQ increases when batches are allowed to be split. To the best of our knowledge, this phenomenon was not pointed out previously. We remark it as a future research challenge to provide formal analysis for this phenomenon and explore how it can be leveraged to improve the performance of token based policies.*

5 Appendix

Proof of Equation (58). To reiterate (58), we prove that

$$\mathbb{E}\left[\left(Q_{l_k}(\tau_{k+1})\right)^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi\right] \leq ([\lambda - \mu_{l_k}]^+)^{1+\gamma} \mathbb{E}[\Delta_k^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi] + C \mathbb{E}[\Delta_k^{0.75(1+\gamma)} \mid \mathcal{F}_{\tau_k}^\xi], \quad (79)$$

where $l_k \in [n]$ is the Last-Idle server during the interval $[\tau_k + 1, \tau_{k+1}]$, $\Delta_k = \tau_{k+1} - \tau_k$ is the duration of the interval and $Q_{l_k}(\tau_{k+1})$ is the queue length at server l_k at time τ_{k+1} . Recall that the RVs l_k , τ_k and $\{Q_i(\tau_k)\}_{i \in [n]}$ are measurable w.r.t $\mathcal{F}_{\tau_k}^\xi$, and that by the definition of the sampling times, we must have $Q(\tau_k) = 0$ and that server l_k receives all incoming jobs during $[\tau_k + 1, \tau_{k+1}]$.

The left hand side of (79) is therefore the $1 + \gamma$ moment of the position, after Δ_k time slots, of a reflected random walk starting at zero. Recall the definition in (57) of $\beta_i := \lambda - \mu_i$ and $\sigma_i^2 := \sigma_a^2 + \sigma_{s_i}^2$. Given the value of l_k , the mean and variance of the step size of this random walk are given by β_{l_k} and $\sigma_{l_k}^2$ respectively.

Lemma 6 of [1] provides an upper bound on the $1 + \gamma$ moment of a very similar reflected random walk. There are two differences between that result and what we want to prove here. First, the random walk in [1] is of the workload process, not the queue length process as we consider here. Specifically, the distribution of the step size has a different dependence on the model parameters. However, the proof in [1] only uses the mean and variance of the step size. Therefore, the same derivation is applicable here and we just plug in our mean and variance.

Second, the bound in [1] is derived for *a fixed number of steps*. In our case, given $\mathcal{F}_{\tau_k}^\xi$, the number of steps is *random*. This is due to the random service capacities. But, the number of steps until the next sampling time is independent of the random walk at server l_k . It only depends on the other servers and how many time slots it takes one of their queue lengths to reach zero.

Thus, a direct application of Lemma 6 in [1] (after plugging in β_{l_k} and σ_{l_k}) yields

$$\mathbb{E}\left[\left(Q_{l_k}(\tau_{k+1})\right)^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi, \Delta_k = t\right] \leq ([\beta_{l_k}]^+)^{1+\gamma} t^{1+\gamma} + C t^{0.75(1+\gamma)},$$

where

$$C = \max_i \{ (16\sigma_i + 8(\sigma_i)^{1/2}(\beta_i)^+)^{(1+\gamma)/2} \}.$$

Thus, we have

$$\begin{aligned} \mathbb{E}\left[\left(Q_{l_k}(\tau_{k+1})\right)^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi\right] &= \sum_{t=1}^{\infty} \mathbb{E}\left[\left(Q_{l_k}(\tau_{k+1})\right)^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi, \Delta_k = t\right] \mathbb{P}(\Delta_k = t \mid \mathcal{F}_{\tau_k}^\xi) \\ &\leq \sum_{t=1}^{\infty} \left(([\beta_{l_k}]^+)^{1+\gamma} t^{1+\gamma} + C t^{0.75(1+\gamma)} \right) \mathbb{P}(\Delta_k = t \mid \mathcal{F}_{\tau_k}^\xi) \\ &= ([\lambda - \mu_{l_k}]^+)^{1+\gamma} \mathbb{E}[\Delta_k^{1+\gamma} \mid \mathcal{F}_{\tau_k}^\xi] + C \mathbb{E}[\Delta_k^{0.75(1+\gamma)} \mid \mathcal{F}_{\tau_k}^\xi]. \end{aligned}$$

□

References

- [1] R. Atar, I. Keslassy, G. Mendelson, A. Orda, S. Vargaftik, Persistent-idle load-distribution, *Stochastic Systems* 10 (2) (2020) 152–169.
- [2] L. Ying, R. Srikant, X. Kang, The power of slightly more than one sample in randomized load balancing, *Mathematics of Operations Research* 42 (3) (2017) 692–722.
- [3] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, A. Greenberg, Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services, *Performance Evaluation* 68 (11) (2011) 1056–1071.

- [4] M. Mitzenmacher, The power of two choices in randomized load balancing, *IEEE Transactions on Parallel and Distributed Systems* 12 (10) (2001) 1094–1104.
- [5] D. Shah, B. Prabhakar, The use of memory in randomized load balancing, in: *Proceedings IEEE International Symposium on Information Theory*, 2002, p. 125.
- [6] S. Asmussen, *Applied probability and queues*, Vol. 51, Springer Science & Business Media, 2008.
- [7] V. A. Malyshev, M. V. Men'shikov, Ergodicity, continuity and analyticity of countable markov chains, *Trudy Moskovskogo Matematicheskogo Obshchestva* 39 (1979) 3–48.
- [8] S. P. Meyn, R. Tweedie, State-dependent criteria for convergence of markov chains, *The Annals of Applied Probability* (1994) 149–168.
- [9] R. Durrett, *Probability: theory and examples*, Vol. 49, Cambridge university press, 2019.