Designing Low-Complexity Heavy-Traffic Delay-Optimal Load Balancing Schemes: Theory to Algorithms

XINGYU ZHOU, The Ohio State University, USA
FEI WU, The Ohio State University, USA
JIAN TAN, The Ohio State University, USA
YIN SUN, Auburn University, USA
NESS SHROFF, The Ohio State University, USA

In this paper, we establish a unified analytical framework for designing load balancing algorithms that can simultaneously achieve low latency, low complexity, and low communication overhead. We first propose a general class Π of load balancing policies and prove that they are throughput optimal and heavy-traffic delay optimal. This class Π includes popular policies such as join-shortest-queue (JSQ) and power-of-d as special cases, but not the recently proposed join-idle-queue (JIQ) policy. In fact, we show that JIQ is not heavy-traffic delay optimal even for homogeneous servers. By exploiting the flexibility offered by the class Π, we design a new load balancing policy called join-below-threshold (JBT-d), in which the arrival jobs are preferably assigned to queues that are no greater than a threshold, and the threshold is updated infrequently. JBT-d has several benefits: (i) JBT-d belongs to the class Π and hence is throughput optimal and heavy-traffic delay optimal. (ii) JBT-d has zero dispatching delay, like JIQ and other pull-based policies, and low message overhead due to infrequent threshold update. (iii) Extensive simulations show that JBT-d has excellent delay performance, comparable to the JSQ policy in various system settings.

CCS Concepts: • **Mathematics of computing** → *Queueing theory*; • **Networks** → *Network performance modeling*; *Network performance analysis*;

Additional Key Words and Phrases: Load balancing; Throughput optimality; Heavy-traffic delay optimality; Sufficient conditions

ACM Reference Format:
Xingyu Zhou, Fei Wu, Jian Tan, Yin Sun, and Ness Shroff. 2017. Designing Low-Complexity Heavy-Traffic Delay-Optimal Load Balancing Schemes: Theory to Algorithms. *Proc. ACM Meas. Anal. Comput. Syst.* 1, 2, Article 39 (December 2017), 30 pages. https://doi.org/10.1145/3154498

1 INTRODUCTION

Load balancing, which is responsible for dispatching jobs on parallel servers, is a key component in computer networks and distributed computing systems. For a large number of practical applications,

This work has been supported in part by NSF grants CNS-1446582, CNS-1719371, CNS-1717060, ONR grant N00014-17-1-2417, ARO grant W911NF-14-1-0368, and a gift from Huawei.
Authors' addresses: Xingyu Zhou, zhou.2055@osu.edu, The Ohio State University, Department of ECE, 2015 Neil Ave. Columbus, OH, 43210, USA; Fei Wu, wu.1973@osu.edu, The Ohio State University, Department of CSE, 2015 Neil Ave. Columbus, OH, 43210, USA; Jian Tan, tan.252@osu.edu, The Ohio State University, Department of ECE, 2015 Neil Ave. Columbus, OH, 43210, USA; Yin Sun, yzs0078@auburn.edu, Auburn University, Department of ECE, 200 Broun Hall, Auburn, AL, 36849, USA; Ness Shroff, shroff.11@osu.edu, The Ohio State University, Departments of ECE and CSE, 2015 Neil Ave. Columbus, OH, 43210, USA.

2476-1249/2017/12-ART39 $15.00
https://doi.org/10.1145/3154498

such as, Web service [6], distributed caching systems (e.g., Memcached [13]), large data stores (e.g., HBase [5]), embarrassingly parallel computations [1] and grid computing [4], the system performance critically depends on the load balancing algorithm it employs.

In a load balancing system, there are two directions of message flows: push messages (from the dispatcher to the servers) and pull messages (from the servers to the dispatcher). In a push-based policy, the dispatcher actively sends query messages to the servers and waits for their responses; In a pull-based policy, the dispatcher passively listens to the report from the servers. The job dispatching decision is conducted at the dispatcher based on the pull-messages sent from the servers. Push-based policies (e.g., the join-shortest-queue (JSQ) policy [19], [2] and the power-of-d policy [10], [17]) have been shown to be delay optimal in the heavy-traffic regime [2], [9]. Recently, the pulled-based policies such as join-idle-queue (JIQ) [8] and the equivalent one in [15], have been proposed. Compared with the push-based policies, these pull-based policies not only achieve good delay performance, but also have some nice features, such as, lower message overhead, lower computational complexity, and zero dispatching delay. However, as shown in the simulations of [8], the delay performance of existing pull-based polices will degrade substantially as the load gets higher. In fact, as shown in Theorem 3.11 of this paper, JIQ is not heavy-traffic delay optimal even for homogeneous servers. Therefore, one key question is how to design load balancing policies that are heavy-traffic delay optimal and meanwhile possess all the nice features of pull-based policies such as zero dispatching delay, low message overhead and low computational complexity.

In this paper, we take a systematic approach to address this question. To that end, the main contributions of this paper are summarized as follows:

- We derive inner-product based sufficient conditions for proving that a load-balancing policy is throughput optimal and heavy-traffic delay optimal. Using these sufficient conditions, we obtain a general class Π of load balancing policies that are throughput optimal and heavy-traffic delay optimal. This class of load balancing policies contains the famous (push-based) JSQ and the power-of-d policies as special cases, but not the (pull-based) JIQ policy.
- On the other hand, we show that JIQ, which is not in Π, is not heavy-traffic delay optimal even for homogeneous servers. While it has been empirically shown in the past that the delay using JIQ is quite bad at high loads, the question of whether it was heavy-traffic delay optimal in homogeneous servers has been previously left unsolved. Furthermore, our novel Lyapunov-drift approach offers a new avenue to show a policy is not heavy-traffic delay optimal.
- By exploiting the significant flexibility offered by class Π, we are able to design a new policy called Join-Below-Threshold (JBT-d). To the best of our knowledge, this is the first load balancing policy that guarantees heavy-traffic delay optimality while enjoying nice features of pull-based policy, e.g., zero dispatching delay, low message overhead and low computational complexity. Through extensive simulations, we demonstrate that JBT-d has excellent delay performance for different system sizes and various arrival and service processes over a large range of traffic loads.

The rest of the paper is organized as follows. Section 1.1 reviews the related work on load balancing schemes. Section 1.2 introduces the necessary notations in the paper. Section 2 describes the system model and the related definitions. Section 3 presents the main results of the paper. In particular, a class Π of flexible load balancing policies are introduced, containing as special cases the popular existing ones and motivating new ones. Sufficient conditions are derived to guarantee throughput and heavy-traffic delay optimality. Section 4 contains the simulation results on comparing different policies, demonstrating the performance and simplicity of our new policy. Section 5 contains the proofs of the main results.

1.1 Related work: push versus pull

This section reviews state-of-the-art load balancing policies with a focus on the system performance in heavy traffic. We group these policies mainly into two categories: push-based and pull-based as shown in Fig. 1.

Push-based policy: Under a push-based policy, the dispatcher tries to "push" jobs to servers. More specifically, upon each job arrival, the dispatcher sends probing messages to the servers, which feed back the required information for dispatching decisions, e.g., queue lengths. After receiving the feedback, the dispatcher sends the incoming jobs to servers based on a dispatching distribution. A classical example in this category is the JSQ policy, under which the dispatcher queries the queue length information of each server upon new job arrivals, and sends the incoming jobs to the server with the shortest queue, with ties broken randomly. It has been shown [19] that for homogeneous servers this policy is delay optimal in a stochastic ordering sense under the assumption of renewal arrival and non-decreasing failure rate service. In the heavy-traffic regime, it has been proved that it is heavy-traffic delay optimal for both heterogeneous and homogeneous servers [2]. Nevertheless, the performance of this policy comes at the cost of substantial overhead as it has to sample the queue lengths of all the servers, which is undesirable in large-scale systems. To overcome this problem, an alternative load balancing policy called power-of-d has been introduced [10], [17]; see also related works [20], [16]. Under this policy, the dispatcher routes all the incoming jobs to the server that has the shortest queue length, with ties broken randomly, out of the d servers sampled uniformly at random. This policy has also been shown to be heavy-traffic optimal for homogeneous servers [9]. However, for heterogeneous servers, the power-of-d policy is neither throughput optimal, nor delay optimal in heavy traffic.

Pull-based policy: Under a pull-based policy, the servers spontaneously send messages to "pull" jobs from the dispatcher according to a fixed policy. One illustrative example is the JIQ policy [8] and the equivalent one in [15]. Under the JIQ policy, each server sends a pull message to the dispatcher whenever it becomes idle. Upon job arrivals, the dispatcher checks the available pull messages in memory. If such messages exist, it removes one uniformly at random, and sends the jobs to the corresponding server. Otherwise, the new jobs will be dispatched uniformly at random to one of the servers in the system. This policy has several favorable properties. The most important property is that the required number of messages in steady-state is at most one for each job arrival, which is smaller than the $2d$ of the power-of-d-choices (d for query and d for response per job). However, as already shown in [8], when the load becomes heavy, the performance of JIQ keeps empirically degrades substantially, and in fact, in Theorem 3.11 we show that it is not heavy traffic delay optimal even for homogeneous servers.

1.2 Notations

We use boldface letters to denote vectors in $\mathbb{R}^N$ and ordinary letters for scalers. Denote by $\overline{\mathbf{Q}}$ the random vector whose probability distribution is the same as the steady-state distribution of $\{\mathbf{Q}(t), t \geq 0\}$. The dot product in $\mathbb{R}^N$ is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{i=1}^N x_i y_i$. For any $\mathbf{x} \in \mathbb{R}^N$, the l_1 norm is denoted by $\|\mathbf{x}\|_1 := \sum_{n=1}^N |x_n|$ and l_2 norm is denoted by $\|\mathbf{x}\| := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. The parallel and perpendicular component of the queue length vector $\mathbf{Q}$ with respect to a vector $\mathbf{c}$ with unit norm is denoted by $\mathbf{Q}_\parallel := \langle \mathbf{c}, \mathbf{Q} \rangle \mathbf{c}$ and $\mathbf{Q}_\perp := \mathbf{Q} - \mathbf{Q}_\parallel$, respectively.

2 MODEL AND DEFINITIONS

This section describes a general model for the load balance systems as shown in Fig. 1, and introduces necessary definitions.

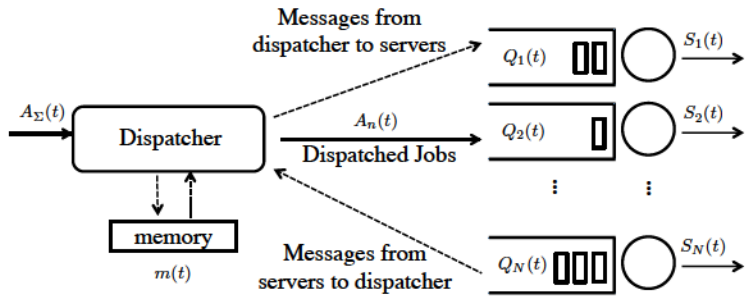

Fig. 1. System model of general load balancing. (a) For push-based policy, we have $m(t) = \emptyset$ for all t since it does not require any memory. The message exchange is bidirectional: probing from the dispatcher and feedback messages from servers. (b) For pull-based policy, $m(t)$ stores the ID of the servers that satisfy a certain condition at time t. The message exchange is unidirectional, i.e., there only exists the pull-message that is sent from the servers to the dispatcher.

2.1 Model Description

Consider a time-slotted load balancing system, with one central dispatcher and N parallel servers. These servers are indexed by its ID $n = 1, 2, \ldots, N$. Each server n is associated with a FIFO (first-in, first-out) queue of length $Q_n(t)$ at the beginning of time slot t, $t = 0, 1, 2, \ldots$. Thus, we use index n to represent both the server and the associated queue. Once a job joins a queue, it will remain in that queue until its service is completed.

Assumption 1 (Arrival Process). Let $A_\Sigma(t)$ and $A_n(t)$ denote the number of exogenous job arrivals and the number of arrivals routed to queue n at time slot t, respectively. We assume that all the exogenous arrivals at time t are routed to one selected queue s, using the standard model as in [2], [9], i.e., $A_s(t) = A_\Sigma(t), s \in \mathcal{N} = \{1, 2, \ldots, N\}$ and $A_i(t) = 0$, for all $i \in \mathcal{N} \setminus \{s\}$. The job arrival process $\{A_\Sigma(t), t \geq 0\}$ is a nonnegative integer valued stochastic process that is *i.i.d* across time t, with mean $\mathbb{E}[A_\Sigma(t)] = \lambda_\Sigma$ and variance $\mathrm{Var}(A_\Sigma(t)) = \sigma_\Sigma^2$. We further assume that the number of exogenous arrivals at each time slot is bounded by a constant, i.e., $A_\Sigma(t) \leq A_{\max} < \infty$ for all $t \geq 0$.

Assumption 2 (Service Process). Let $S_n(t)$ denote the potential service offered to queue n at time t, which represents the maximum number of jobs that can be served in time slot t. Therefore, if the offered service $S_n(t)$ is larger than the number of pending jobs in queue n at time slot t, it will cause an unused service $U_n(t)$, as defined in (1). For each n, the process $\{S_n(t), t \geq 0\}$ is a nonnegative integer valued *i.i.d.* stochastic process with mean $\mathbb{E}[S_n(t)] = \mu_n$ and variance $\mathrm{Var}(S_n(t)) = \nu_n^2$. Moreover, $\lambda_\Sigma < \sum_{i=1}^N \mu_n$. Furthermore, the processes $\{S_n(t), t \geq 0\}, n \in \mathcal{N}$ are mutually independent across different queues, which are also independent of the arrival processes. The offered service $S_n(t)$ to each queue is uniformly bounded by a constant, i.e., $S_n(t) \leq S_{\max} < \infty$ for all $t \geq 0$ and all $n \in \mathcal{N}$.

Let $\mathrm{Q}(t) = \{Q_1(t), \ldots, Q_N(t)\}$ be the queue lengths observed at the beginning of time t. Define $m(t)$ to be the set of server IDs stored in the dispatcher at the beginning of time slot t. In general, the dispatcher makes the decision of $A_n(t)$ based on $(\mathrm{Q}(t), m(t))$ for each time slot t. This includes the cases that the dispatching decision depends only on $\mathrm{Q}(t)$ (e.g., JSQ), partial information of $\mathrm{Q}(t)$ (e.g., power-of-d) or only on $m(t)$ (e.g., JIQ). In each time slot, the queueing dynamics evolves according to the following procedure. The job arrivals occur at the beginning of time slot t. Then, the dispatching decision $A_n(t)$ is selected based on $(\mathrm{Q}(t), m(t))$. Further, the routed jobs are processed by the allocated servers. Thus, the queueing dynamics is given by the following equation,

$$\begin{aligned} Q_n(t+1) &= [Q_n(t) + A_n(t) - S_n(t)] \\ &= Q_n(t) + A_n(t) - S_n(t) + U_n(t), \end{aligned} \tag{1}$$

where $[x]^+ = \max(0, x)$, $U_n(t) = \max(S_n(t) - Q_n(t) - A_n(t), 0)$ denotes the unused service of queue n.

2.2 Definitions

The load balancing system is modeled as a discrete-time Markov chain $\{Z(t) = (\mathrm{Q}(t), m(t)), t \geq 0\}$ with state space $\mathcal{Z}$, using queue length vector $\mathrm{Q}(t)$ together with the memory state $m(t)$. We consider a system $\{Z^{(\epsilon)}(t), t \geq 0\}$ parameterized by ϵ, i.e., the exogenous arrival process is $\{A_\Sigma^{(\epsilon)}(t), t \geq 0\}$ with $\lambda_\Sigma^{(\epsilon)} = \mu_\Sigma - \epsilon = \sum_n \mu_n - \epsilon$. That is, we use ϵ to indicate the distance of arrival rate to the capacity boundary, and it is also adopted as a superscript to represent the corresponding random variables and processes.

Definition 2.1 (Stability). $\{Z^{(\epsilon)}(t), t \geq 0\}$ is said to be stable if we have

$$\limsup_{C\to\infty} \limsup_{t\to\infty} \mathbb{P}\left(\sum_n Q_n^{(\epsilon)}(t) > C\right) = 0.$$

A load balancing policy is said to be throughput optimal if it stabilizes the system under any arrival rate in the capacity region. Since the capacity region in our model is simply $\lambda_\Sigma < \mu_\Sigma$, the definition of throughput optimality is given as follows.

Definition 2.2 (Throughput Optimality). A load balancing policy is said to be throughput optimal if it stabilizes $\{Z^{(\epsilon)}(t), t \geq 0\}$ for any $\epsilon > 0$.

For the definition of heavy-traffic delay optimality, we need the following definition and property.

Definition 2.3 (Resource-pooled System). A single-server FCFS (first-come, first-serve) system $\{q^{(\epsilon)}(t), t \geq 0\}$ is said to be the resource-pooled system with respect to $\{Z^{(\epsilon)}(t), t \geq 0\}$, if its arrival and service process satisfy $a^{(\epsilon)}(t) = A_\Sigma^{(\epsilon)}(t)$ and $s(t) = \sum S_n(t)$ for all $t \geq 0$. Then, we have

$$\mathbb{E}\left[q^{(\epsilon)}(t)\right] \leq \mathbb{E}\left[\sum Q_n^{(\epsilon)}(t)\right], \tag{2}$$

for all $t \geq 0$ and $\epsilon > 0$.

In words, a resource-pooled system is a system that merges the total resource of N servers and queues to a single server with a single queue. Eq. (2) holds due to the fact for any t, the overall arrivals to the resource-pooled system and to load balancing system are the same, and the overall service in the resource-pooled system is stochastically larger than the overall service in the load balancing system. This is due to the fact that the jobs in load balancing system cannot be moved from one queue to another, which often results in a strict inequality in Eq. (2). However, in the heavy-traffic regime, this lower bound can be achieved under some policy in an asymptotic sense as defined in the next definition, and hence based on Little's law this policy achieves the minimum average delay of the system.

Definition 2.4 (Heavy-traffic Delay Optimality). A load balancing policy is said to be heavy-traffic delay optimal if the stationary workload of $\{Z^{(\epsilon)}(t), t \geq 0\}$ under all the arrival and service processes in Assumptions 1 and 2, satisfies [1]

[1] Assume $(\sigma_\Sigma^{(\epsilon)})^2$ converges to a constant.

$$\lim_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\sum_n \overline{Q}_n^{(\epsilon)}\right] = \lim_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\overline{q}^{(\epsilon)}\right], \tag{3}$$

where $\overline{\mathrm{Q}}$ is the random vector whose probability distribution is the same as the steady-state distribution of $\{\mathrm{Q}(t), t \geq 0\}$.

Remark 1. Based on the definition above, in order to show a policy, say $\mathcal{P}_1$, is not heavy-traffic delay optimal, it is sufficient to find a class of $\{A_\Sigma^{(\epsilon)}(t)\}$ and $\{S_n(t)\}$ such that Eq. (3) does not hold. In other words, there exists a class of arrival and service processes for which policy $\mathcal{P}_1$ cannot achieve the lower bound (i.e., the resource-pooled system) while JSQ can (since it is heavy-traffic delay optimal).

3 MAIN RESULTS

In this section, we introduce a class Π of load balancing policies which are proven to be delay-optimal in the heavy-traffic regime. Popular load balancing policies, such as JSQ and power-of-d, are special cases in Π; but the JIQ policy does not belong to Π as we will show in Theorem 3.11 that it is not heavy-traffic delay optimal. In order to improve the delay performance of JIQ while maintaining its low message overhead and simplicity, we develop a new load balancing policy named join-below-threshold (JBT-d), which is heavy-traffic delay-optimal as we can show JBT-d is in Π and has a low message overhead similar to JIQ.

3.1 The Class of Load Balancing Policies Π

Let us denote $\mathrm{p}(t) = (p_1(t), \ldots, p_N(t))$, where $p_n(t)$ is the probability that the new arrivals in time slot t are dispatched to queue n such that $\sum_{n=1}^N p_n(t) = 1$. We consider a class of load balancing policies in which $\mathrm{p}(t)$ is a function of the system state $Z(t) = \{\mathrm{Q}(t), m(t)\}$. Consider a permutation $\sigma_t(\cdot)$ of $(1, 2, \ldots, N)$ that satisfies $Q_{\sigma_t(1)}(t) \leq Q_{\sigma_t(2)}(t) \leq \ldots \leq Q_{\sigma_t(N)}(t)$ for all t, i.e., the queues are sorted according to an increasing order of the queue lengths in time slot t with ties broken randomly. Define $\mathrm{P}(t) = (P_1(t), \ldots, P_N(t))$ such that $\mathrm{P}(t)$ is a permutation of $\mathrm{p}(t)$ with $P_n(t) = p_{\sigma_t(n)}(t)$. Let

$$\begin{aligned} \Delta_n(t) &= p_{\sigma_t(n)}(t) - \mu_{\sigma_t(n)}/\mu_\Sigma \\ &= P_n(t) - \mu_{\sigma_t(n)}/\mu_\Sigma. \end{aligned} \tag{4}$$

Definition 3.1 (Equivalence in inner-product). A dispatching distribution $\hat{\mathrm{P}}(t)$ is said to be *equivalent to another dispatching distribution* $\mathrm{P}(t)$ *in inner product*, if

$$\sum_n Q_{\sigma_t(n)} \Delta_n(t) = \sum_n Q_{\sigma_t(n)} \hat{\Delta}_n(t), \tag{5}$$

or equivalently, if

$$\sum_n Q_{\sigma_t(n)} P_n(t) = \sum_n Q_{\sigma_t(n)} \hat{P}_n(t). \tag{6}$$

The equivalence between (5) and (6) follows immediately from (4). Intuitively speaking, a load-balancing policy is 'good' if the inner product between $\mathrm{Q}_{\sigma_t}(t)$ and $\mathrm{P}(t)$ is as small as possible such that more packets are dispatched to shorter queues. If $\mathrm{P}(t)$ is equivalent to $\hat{\mathrm{P}}(t)$ in inner-product, we can replace $\hat{\mathrm{P}}(t)$ by $\mathrm{P}(t)$ without affecting the property of the policy in heavy-traffic regime, which will be explained in details later.

The following definitions enable us to distinguish different load balancing policies based on $\mathrm{P}(t)$ or equivalently $\Delta(t)$:

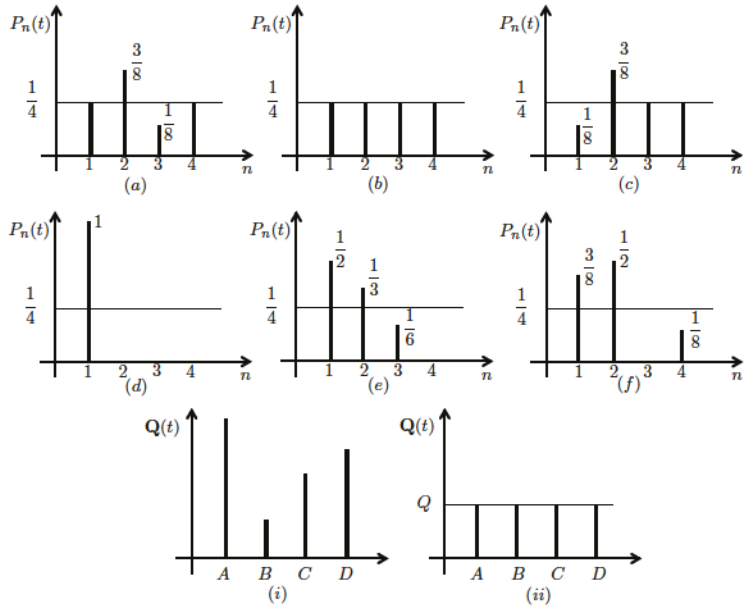

Fig. 2. Illustrations of tilted distribution, δ-tilted distribution, and equivalence in inner-product.

Definition 3.2 (Tilted distribution). A dispatching distribution $\mathrm{P}(t)$ is said to be *tilted*, if there exists $k \in \{2, \ldots, N\}$ such that $\Delta_n(t) \geq 0$ for all $n < k$ and $\Delta_n(t) \leq 0$ for all $n \geq k$.

Definition 3.3 (δ-tilted distribution). A dispatching distribution $\mathrm{P}(t)$ is said to be *δ-tilted*, if (i) $\mathrm{P}(t)$ is tilted and (ii) these exists a constant $\delta > 0$ such that $\Delta_1(t) \geq \delta$ and $\Delta_N(t) \leq -\delta$.

Some examples are presented in Fig. 2 to facilitate the understanding of tilted distribution, δ-tilted distribution, and equivalence in inner-product. Fig. 2 (a)-(f) illustrate six dispatching distributions $\mathrm{P}(t)$. The queue state $\mathrm{Q}(t)$ is given by (i) or (ii). The service rates are $\mu_A = \mu_B = \mu_C = \mu_D = 1$ such that $\mu_i/\mu_\Sigma = 1/4$ for $i = A, B, C, D$. By direct computation, one can obtain that $P_n(t)$ is tilted in scenario (a), (b), (d), (e), and (f), and is δ-tilted in scenario (d), (e), and (f). If $\mathrm{Q}(t)$ is in the State (i), there is no tie in the queue length and hence the permutation $\sigma_t(\cdot)$ is unique, which means that $\mathrm{P}(t)$ is fully determined by $\mathrm{p}(t)$. If $\mathrm{Q}(t)$ is in the State (ii), all queue lengths are equal and hence the permutation $\sigma_t(\cdot)$ is non-unique, which means that $\mathrm{P}(t)$ is determined by both $\mathrm{p}(t)$ and $\sigma_t(\cdot)$. In this case, however, the inner product between $\mathrm{Q}_{\sigma_t}(t)$ and $\mathrm{P}(t)$ is 1 in all (a)-(f), and hence the dispatching distributions $\mathrm{P}(t)$ in (a)-(f) are mutually equivalent in inner product. For example, in this case even though $\mathrm{P}(t)$ in (c) is neither tilted nor δ-tilted, it is equivalent in inner product to $\mathrm{P}(t)$ in (d) which is both tilted and δ-tilted.

From the perspective of heavy-traffic delay performance, tilted distribution is a dispatching distribution that is not worse than random routing and δ-tilted distribution is a dispatching distribution that is strictly better than random routing. In addition, the equivalence in inner-product allows us to transfer a tilted dispatching distribution to a δ-tilted dispatching distribution when there are ties in queue lengths, that is, it allows to merges probability in $\mathrm{P}(t)$ from longer queues to shorter queues without changing the inner product.

We now introduce a class of load balancing algorithms Π based on the property of $\mathrm{P}(t)$ or its equivalent distributions in inner product.

Definition 3.4. A load balancing policy is said to belong to class Π if it satisfies the following two conditions:

(i) $\mathrm{P}(t)$ or one of its equivalent distributions in inner product is tilted for all $Z(t)$ and $t \geq 0$.
(ii) For some finite positive constants T and δ that both are independent of ϵ, there exists a time slot $t_k \in \{kT, kT+1, \ldots, (k+1)T-1\}$ for each $k \in \mathbb{N}$ such that $\mathrm{P}(t_k)$ or one of its equivalent distributions in inner product is δ-tilted for all $Z(t_k)$.

In the sequel, we will show that any policy in Π satisfies the following two sufficient conditions for throughput and heavy-traffic delay optimality, which are obtained via the Lyapunov-drift based approach developed in [2].

Lemma 3.5. *If there exist $T_1 > 0$, $K_1 \geq 0$, and $\gamma > 0$ such that for all $t_0 = 1, 2, \ldots$, all $Z \in \mathcal{Z}$ and $\lambda_\Sigma < \mu_\Sigma$*

$$\mathbb{E}\left[\sum_{t=t_0}^{t_0+T_1-1} \langle \mathrm{Q}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t_0) = Z\right] \leq -\gamma \|\mathrm{Q}\| + K_1, \tag{7}$$

then the system is throughput-optimal. Moreover, the stationary distribution of the queueing system has bounded moments, i.e., there exist finite M_r such that for all $\epsilon > 0$ and $r \in \mathbb{N}$

$$\mathbb{E}\left[\left\|\overline{\mathrm{Q}}^{(\epsilon)}\right\|^r\right] \leq M_r.$$

Proof. See Appendix A. □

Lemma 3.6. *Under the assumptions of Lemma 3.5, if there further exist $T_2 > 0, K_2 \geq 0$ and $\eta > 0$ that are independent of ϵ, such that for all $t_0 = 1, 2, \ldots$ and all $Z \in \mathcal{Z}$*

$$\mathbb{E}\left[\sum_{t=t_0}^{t_0+T_2-1} \langle \mathrm{Q}_\perp(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t_0) = Z\right] \leq -\eta \|\mathrm{Q}_\perp\| + K_2 \tag{8}$$

holds for all $\epsilon \in (0, \epsilon_0)$, $\epsilon_0 > 0$, where $\mathrm{Q}_\perp = \mathrm{Q} - \langle \mathrm{Q}, \mathrm{c}\rangle \mathrm{c}$ is the perpendicular component of Q with respect to the line $\mathrm{c} = \frac{1}{\sqrt{N}}(1, 1, \ldots, 1)$, then the system is heavy-traffic delay optimal, i.e.,

$$\lim_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\sum_n \overline{Q}_n^{(\epsilon)}\right] = \lim_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\overline{q}^{(\epsilon)}\right].$$

Proof. See Appendix B. □

Remark 2. Note that these two sufficient conditions distilled from the Lyapunov-drift based approach not only provide a unified approach for throughput and heavy-traffic optimality analysis, but also enable us to abstract a class of heavy-traffic delay optimal policies. In particular, using Lemma 3.5 and Lemma 3.6, we are able to prove the main result of this paper.

Theorem 3.7. *Any load balancing policy in Π is throughput optimal and heavy-traffic delay optimal.*

Proof sketch of Theorem 3.7. The insight for a policy in Π to satisfy the sufficient condition in Eq. (7) is that under tilted dispatching distribution the performance is no worse than random dispatching. This follows from the following property of tilted distribution

$$\sum_{n=1}^{N} Q_{\sigma_t(n)}(t) \Delta_n(t) \leq 0. \tag{9}$$

The equality is obtained when all $\Delta_n(t)$ is zero, which is the case of random dispatching as shown in (b) of Fig. 2. Note that for all other cases of a tilted distribution, Eq. (9) is strictly less than zero. This is true since $\sum_{n=1}^{N} \Delta_n(t)$ is always zero and the permutation is in the non-decreasing order of the queue length.

The intuition for a policy in Π to satisfy the sufficient condition in Eq. (8) is that the performance under any δ-tilted dispatching distribution is strictly better than random dispatching, under which the term in Eq. (8) is 0 for homogeneous servers and of order ϵ for heterogeneous servers. Note that under a δ-tilted distribution, we have

$$\sum_{n=1}^{N} Q_{\sigma_t(n)}(t)\Delta_n(t) \leq -\delta(Q_{\sigma_t(N)}(t) - Q_{\sigma_t(1)}(t). \tag{10}$$

This inequality comes from the definition of the δ-tilted distribution and fact that the permutation is in the non-decreasing order of the queue length. In order to have the term of $\|\mathrm{Q}_\perp\|$, the following inequality would be quite useful

$$\|\mathrm{Q}_\perp(t)\| \leq \sqrt{N}(Q_{\sigma_t(N)}(t) - Q_{\sigma_t(1)}(t)). \tag{11}$$

This is true since $\mathrm{Q}_\perp(t) = \mathrm{Q}(t) - \mathrm{Q}_\parallel(t) = \mathrm{Q}(t) - \frac{\sum Q_n(t)}{N}\mathbf{1} = \mathrm{Q}(t) - Q_{\text{avg}}(t)\mathbf{1}$, in which $Q_{\text{avg}}(t)$ is the average queue length among the N servers at time slot t.

The details of the proof are presented in Section 5.1. □

From Eqs. (9) and (10), it can be seen that the important property of a given policy is fully characterized by the inner product of $\mathrm{Q}_{\sigma_t}(t)$ and $\Delta(t)$ under the system state $Z(t)$, which is actually the motivation to define equivalent distribution in inner product. That is, even though the dispatching distribution $\mathrm{P}(t)$ is not unique when there are ties in queue lengths, the inner product is actually the same if two dispatching distributions are equivalent in inner product, hence preserving the same property in heavy-traffic regime.

Note that class Π is sufficient but not necessary for heavy-traffic delay optimality. Nevertheless, in the next section, we will show that it not only contains many well-known heavy-traffic delay optimal policies but also allows us to design new heavy-traffic delay optimal policies which enjoy nice features of pull-based policies.

3.2 Important Policies in Π

3.2.1 Join-shortest-queue (JSQ) policy. Under JSQ policy, all the incoming jobs are dispatched to the queue that has the shortest queue length, ties are broken uniformly at random, out of all the servers.

Proposition 3.8. *The JSQ policy belongs to* Π*, and hence is throughput optimal and heavy-traffic delay optimal.*

The result that JSQ is throughput and heavy traffic delay optimal has been first proven via diffusion limits for two servers in [3] and via Lyapunov-drift argument for N servers in [2]. Here, we present another simple proof based on our main result.

Proof. Note that when there are no ties in queue lengths, the dispatching distribution $\mathrm{P}(t)$ under JSQ satisfies that for all t

$$P_1(t) = 1 \text{ and } P_n(t) = 0, 2 \leq n \leq N. \tag{12}$$

In other words, all the arrivals are dispatched to the shortest queue, which is always the queue $\sigma_t(1)$ if there are no ties in queue lengths. If there are ties in queue lengths, this $\mathrm{P}(t)$ is equivalent in inner product to other dispatching distribution under the state $Z(t)$ in which ties exist. In particular,

if there are $m \leq N$ queues that all have the shortest queue length, then in this case by random routing the dispatching distribution under JSQ is given by $\hat{P}_i = \frac{1}{m}$ for all $1 \leq i \leq m$, and $\hat{P}_i = 0$ for all $i > m$. It can be seen that $\mathrm{P}(t)$ in Eq. (12) is equivalent in inner product to $\hat{\mathrm{P}}(t)$ according to the definition because $Q_{\sigma_t(1)} = Q_{\sigma_t(2)} = \ldots = Q_{\sigma_t(m)}$. Thus, for all $Z(t)$, under JSQ the dispatching distribution or its equivalent distribution in inner product is in the form of Eq. (12). Hence, we have $\Delta_1(t) = 1 - \mu_{\sigma_t(1)}/\mu_\Sigma > 0$, and $\Delta_n(t) = -\mu_{\sigma_t(n)}/\mu_\Sigma < 0$ for all $2 \leq n \leq N$, which implies that $\mathrm{P}(t)$ is a δ-tilted distribution with $\delta = \mu_{min}/\mu_\Sigma$ for all $Z(t), t \geq 0$, where $\mu_{min} = \min_{n \in \mathcal{N}} \mu_n$. Therefore, the JSQ policy is contained in the class Π under both heterogeneous and homogeneous servers. □

3.2.2 The power-of-d policy. Under the power-of-d policy, all the incoming jobs are dispatched to the queue that has the shortest queue length, ties are broken uniformly at random, out of $d \geq 2$ servers, which are chosen uniformly at random.

Proposition 3.9. *The power-of-d policy belongs to* Π *under homogeneous servers, and hence is throughput-optimal and heavy-traffic delay optimal.*

The power-of-d policy has been proven to be heavy-traffic delay optimal via Lyapunov drift condition in [9]. Here, we will present another proof based on our main result.

Proof. Note that when there are no ties in queue lengths, the dispatching distribution $\mathrm{P}(t)$ under the power-of-d policy satisfies that for all $t \geq 0$

$$P_n(t) = \binom{N-n}{d-1} \Big/ \binom{N}{d}, 1 \leq n \leq N-d+1, \tag{13}$$

and $P_n(t) = 0$, for all $n > N - d + 1$. This comes from the fact that all arrivals are dispatched to the queue with shortest queue length among d uniformly randomly sampled servers. Thus, if the queue $\sigma_t(n)$ is the one with shortest queue length among d samples, the remaining $d-1$ samples must come from queues $\sigma_t(n+1), \sigma_t(n+2), \ldots \sigma_t(N)$ if all the queue lengths are different in $Z(t)$. If there are ties in queue lengths, it can be easily shown that this $\mathrm{P}(t)$ is equivalent in inner product to other dispatching distributions under any given $Z(t)$ in which there are ties in queue lengths. Thus, for all $Z(t)$, the dispatching distribution or its equivalent distribution in inner product under the power-of-d policy can be fully determined by Eq. (13). Since $P_n(t)$ is decreasing and $\mu_{\sigma_t(n)} = \mu$ under homogeneous servers, $\mathrm{P}(t)$ is a tilted distribution. Note that $\Delta_1(t) = \frac{d-1}{N}$ and $\Delta_N(t) = -\frac{1}{N}$. As a result, $\mathrm{P}(t)$ is a δ-tilted distribution with $\delta = \frac{1}{N}$ for all $Z(t)$, which implies that power-of-d policy is included in the class Π for homogeneous servers. □

3.2.3 Join-idle-queue policy is not in Π. Now we will show that the JIQ policy is not contained in the class Π because it is in fact not heavy-traffic delay optimal in homogeneous servers. For the heterogeneous case, it is well-known that JIQ is not heavy-traffic delay optimal since it is not even throughput optimal for a fixed number of servers [15]. However, for the homogeneous case, it is still open whether it is heavy-traffic optimal for a fixed number of servers, although it has been shown to be heavy-traffic optimal when the number of servers goes to infinity in the Halfin-Whitt regime [12]. It turns out that when the number of servers is fixed, there exists a class of arrival process, under which the delay performance of JSQ is strictly better than that of JIQ in the heavy-traffic limit. More specifically, as shown in the proof of Theorem 3.11, for a class of arrival process, the delay under JIQ cannot achieve the common lower bound (i.e., the resource-pooled system), while JSQ can, which implies that JIQ is not heavy-traffic delay optimal for homogeneous case.

In particular, we consider the two-server case with constant service process with rate 1. We are able to find a class of arrival process such that Eq. (3) under JIQ does not hold. Let us first introduce the class of arrival process $\mathcal{A}$.

Table 1. Summary of load balancing policies

Policy	Message	Throughput-Optimal		Heavy-traffic Delay-Optimal	
		Homogeneous	Heterogeneous	Homogeneous	Heterogeneous
Random	0	√	×	×	×
JSQ [2]	$2N$	√	√	√	√
Power-of-d [9], [10]	$2d$	√	×	√	×
JIQ[15], [8]	≤ 1	√	×	×	×
JBT-d	$\leq \frac{N+2d}{T}+1$	√	×	√	×
JBTG-d	$\leq \frac{N+2d}{T}+1$	√	√	√	√

*The message rate for JBT-d and JBTG-d in this table is just a crude upper bound. When the new threshold is larger than the old one, there is no need for the servers that are already recorded in memory to resend pull-messages.

Definition 3.10. An arrival process $A_\Sigma(t)$ is said to belong to $\mathcal{A}$ if

(i) $\mathbb{P}(A_\Sigma^{(\epsilon)}(t)=0)=p_0$, which p_0 is a constant independent of ϵ.

(ii) $(\sigma_\Sigma^{(\epsilon)})^2$ approaches a constant σ_Σ^2 which satisfies that $\sigma_\Sigma^2 > 8/p_0 - 4$.

More concretely, we are able to show the following result.

THEOREM 3.11. *JIQ is not heavy-traffic delay optimal in a load balancing system consisting of two homogeneous servers.*

PROOF. The proof is relegated to the technical report [21]. □

3.3 Designing New Policies in Π

It has been shown in the last section that the state-of-art push-based policies, e.g., JSQ and power-of-d, are all included in Π. Recall that, both of them need to sample the queue length information upon each new arrival, which directly results in the following two problems.

(a) The message exchange rate between dispatcher and servers is high, especially for join-shortest-queue.

(b) Each arrival has to wait for completion of the message exchange before being dispatched, which increases the actual response time for each job.

To resolve the problem, the pull-based policies, join-idle-queue (JIQ) in [8] and an equivalent algorithm called PULL in [15] are proposed, which have been shown to enjoy low message rate (at most one message per job) and have a better performance than the power-of-d policy from light to moderate loads. However, as shown via numerical results in [8] and the proof of Theorem 3.11 in this paper, when the load becomes high, the performance of JIQ is much worse than the power-of-d policy, which motivates us to design policies that enjoy low message rates, while still guaranteeing throughput and heavy-traffic delay optimality.

Definition 3.12. Join-below-threshold-d (JBT-d) policy is composed of three components:

(1) A threshold is updated every T units of time by uniformly at random sampling d servers, and taking the shortest queue length among the d servers as the new threshold.

(2) Each server sends its ID to the dispatcher when its queue length is not larger than the threshold for the first time.

(3) Upon a new arrival, the dispatcher checks the available IDs in the memory. If they exist, it removes one uniformly at random, and sends all the new arrivals to the corresponding server. Otherwise, all the new arrivals will be dispatched uniformly at random to one of the servers in the system.

To be more specific, we explain the connections of the three components as follows. At the beginning of each time slot, the dispatcher immediately routes the new arrivals to a server only based on its memory state, i.e., no sampling. If there are available IDs in memory, it removes one uniformly at random and sends the newly arrived jobs to the corresponding server. Otherwise, it sends the new jobs to a server selected uniformly at random among all the servers. At the end of each time slot, if there is no update of threshold, each server will immediately report its ID if its queue length is not larger than the threshold for the first time, i.e., only reporting once for each server before dispatched. Otherwise, the dispatcher updates the threshold by uniformly at random sampling d servers, and the new threshold is set as the shortest queue length among d samples. Then, each server decides to whether or not to report based on its queue length and the new threshold, using the same way as before.

Definition 3.13. The JBT-d policy can be easily generalized for heterogeneous servers, denote by JBTG-d, as follows. The only difference is that the dispatching probability distribution for the case of non-empty and empty memory is given by

$$\psi_i(t) := \frac{\mu_i}{\sum_{j\in m(t)}\mu_j}\mathbb{1}_{\{i\in m(t)\}} \text{ and } \phi_i(t) := \frac{\mu_i}{\mu_\Sigma} \text{ for all } i.$$

That is, the probability to be selected for a server that has its ID in memory is weighted by its service rate. This can be easily done by requiring the server to report its service rate μ_n as well as its ID.

In the following, we will show that JBT-d and JBTG-d belong to Π, and hence throughput and heavy-traffic delay optimal. More specifically, we have the following result.

Proposition 3.14. *For any finite T and $d \geq 1$, the following two assertions are true:*

(1) JBT-d is in Π for homogeneous servers, and hence throughput and heavy-traffic delay optimal.
(2) JBTG-d is in Π for both homogeneous and heterogeneous servers, and hence throughput and heavy-traffic delay optimal.

Proof sketch of Proposition 3.14. Let us look at JBT-d for some key insights behind this proof. In order to show it is in Π, we only need to show that it satisfies the two conditions (i) and (ii). For the condition (i), we will show that at any time slot t, the dispatching is no worse than the random routing. For the condition (ii), we will show that at time slots $rT+1$, $r \in \{0, 1, 2, \ldots\}$, the dispatching decision is strictly better than the random routing.

Note that under the JBT-d policy, if the ID of the server $\sigma_t(n+1)$ is in $m(t)$, we must have that the ID of the server $\sigma_t(n)$ is also in $m(t)$ as the permutation is in the non-decreasing order of the queue length. Denote by $\widetilde{p}_k(t)$ the probability that there are k IDs in the memory $m(t)$ for time t, i.e., $\widetilde{p}_k(t) = \Pr(|m(t)| = k)$. Then, the probability for the server $\sigma_t(n)$ to be selected at time t, i.e., $P_n(t)$ is given by

$$P_n(t) = \sum_{i=n}^{N} \widetilde{p}_i(t)\frac{1}{i}. \tag{14}$$

This is true since for the server $\sigma_t(n)$ to be selected, there should be at least n IDs in memory, i.e., $|m(t)| \geq n$ and in each case the probability for the server $\sigma_t(n)$ to be chosen is $\frac{1}{|m(t)|}$. Therefore, we can see that the probability of $P_n(t)$ satisfies

$$P_1(t) \geq P_2(t) \geq \ldots \geq P_N(t), \tag{15}$$

which directly implies that for all $t \geq 0$ there exists a k between 2 and N such that $\Delta_n(t) = P_n(t) - \frac{1}{N} \geq 0$ for all $n < k$ and $\Delta_n(t) \leq 0$ for all $n \geq k$. Therefore, condition (i) of Π is satisfied.

For condition (ii), we will show that there exists a lower bound for δ such that $\mathrm{P}(rT+1)$ (or an inner product equivalent distribution when there are ties in queue lengths), is at least a δ-tilted distribution for $r \in \{0, 1, 2, \ldots\}$. In this case, we need only to show that $P_N(rT+1)$ is strictly less than $\frac{1}{N}$ for all the system state $Z(rT+1)$.

The full proof is presented in Section 5.2. □

3.4 Features of JBT-d

This section summarizes the main features of JBT-d policy and compares it with existing policies in Table 1. In particular, we compare the number of messages for each new arrival under different policies. For push-based polices, e.g., JSQ and power-of-d, there are d query and d response messages for each new arrival ($d = N$ for JSQ policy). For JIQ policy, for each new arrival, it requires at most one pull-message since when there are no pull-messages in memory, the arrival is dispatched randomly without costing any pull-message. Similarly, our JBT-d policy requires $2d$ push-messages every T time slots to update the threshold. Due to the threshold update, the old pull-messages may be discarded, which is upper bounded by N. Hence, the pull-message for each new arrival under JBT-d is at most $1 + \frac{2d+N}{T}$.

In sum, the JBT-d policy has the following nice features: a) It is throughput and heavy-traffic delay optimal since it is in Π. b) It is able to guarantee heavy-traffic delay optimal with very low message overhead when T is relatively large. c) The computation overhead is small since it only needs to keep a list of the available IDs and choose randomly. d) The arrival is immediately dispatched, i.e., there is no dispatching delay as compared to push-based policies such as JSQ and Power-of-d.

It is worth pointing that by just changing the way of updating the threshold in JBT-d, we can design other new policies which also enjoy the nice features above. For example, it can be easily shown via similar arguments that if the threshold is updated by sampling all the servers and taking the average value of the queue length as the new threshold, this corresponding new policy is still in the class Π.

4 NUMERICAL RESULTS

In this section, we use simulations to compare our proposed policies JBT-d and JBTG-d with join-shortest-queue (JSQ), join-idle-queue (JIQ), power-of-d (SQ(d)) and power-of-d with memory (SQ(d,m)). The power-of-d with memory policy (SQ(d,m)) improves power-of-d by using extra memory to store the m shortest queues sampled at the previous time slot [11].

We compare the throughput performance, delay performance, heavy-traffic delay performance and message overhead performance under various arrival and service processes as well as different system sizes. Moreover, the 95% confidence intervals for all the simulation results can be found in the technical report [21], which justify the accuracy of the simulation results. The exogenous arrival $A_\Sigma(t)$ and potential service $S_n(t)$ are drawn from a Poisson distribution with rate λ_Σ and μ_n for each time slot unless otherwise specified. The traffic load is equal to $\rho = \lambda_\Sigma/\mu_\Sigma$. The parameter T is the threshold update interval for JBT-d and JBTG-d.

Below we summarize the key observations from the simulations; see Appendix D of the technical report [21] for the full set of simulation results.

(i) **Throughput performance:**
   (a) Our proposed policy JBT-d stabilizes all the considered loads in heterogeneous systems under all different settings.
   (b) JIQ and SQ(d) cannot stabilize the system when the load is high in all the cases.
   (c) JIQ appears to have a larger capacity region as the number of servers increases. This agrees with the theoretical result in [15].

(ii) **Delay performance:**
  (a) Our proposed policy JBT-d exhibits good performance across a wide range from light to heavy traffic in all the cases.
  (b) As the system size increases, JBT-d achieves the same performance as JSQ for a larger range of loads. Meanwhile, the gains of JBT-d over SQ(d) and SQ(d,m) become larger as the number of servers increases.
  (c) The gain of JBT-d over JIQ decreases as the number of servers increases. This is also intuitive since as N goes to infinity, it is more likely to find an idle server, which results in the fact that JIQ is heavy-traffic delay optimal in the Halfin-Whitt regime [12].
  (d) The gain of JBT-d over JIQ increases as the arrivals or services become more bursty. This agrees with the insight in the proof of Theorem 3.11 that larger variance of arrival or service process will degrade the performance of JIQ.

(iii) **Message overhead performance:**
  (a) Our proposed policy JBT-d continues to have a low message overhead among all the cases.
  (b) Push-based policies such as SQ(d) and SQ(d,m) have to increase their message overhead linearly with respect to d to achieve good delay performance as the system size increases. In contrast, our proposed JBT-d is able to achieve good performance with a message rate that is less than 1 for all the cases when T is large.

(iv) **Confidence interval:**
  (a) The 95% confidence intervals of the response time under JBT-d is small for all the various settings as shown in the following figures and the additional results in Appendix E of the technical report [21].

Next, we will provide details for the three metrics on throughput, delay and message overhead, respectively.

4.1 Throughput Performance

We investigate the throughput region of different load balancing policies in the case of heterogeneous servers. In particular, we consider the case that the system consisting of two server pools each with five servers and the rates are 1 and 10, respectively. A turning point in the curve indicates that the load approaches the throughput region boundary of the corresponding policy.

Figure 3 shows that the system becomes unstable when $\rho > 0.5$ under the policy power-of-2 (SQ(2)), and it becomes unstable under JIQ when $\rho > 0.9$. In contrast, our proposed JBTG-d policy remains stable for all the considered loads which agrees with the theoretical results. It can be seen that JBT-2 is also able to stabilize the system for all the considered loads in this case. Note that the system remains stable under the power-of-2 with memory policy SQ(2,3), which demonstrates the benefit of using memory to obtain maximum throughput as first discussed in [14].

We further provide additional simulation results on throughput performance under different arrival and service process as well as different system sizes in the technical report [21].

4.2 Delay Performance

We investigate the mean response time under different load balancing policies in homogeneous servers with different system sizes and various arrival and service processes. The time interval for threshold update of JBT-d is set $T = 1000$.

Let us first look at the regime when ρ is from 0.3 to 0.99, which ranges from light traffic to heavy traffic. Figure 4 shows that our proposed policy JBT-d outperforms both power-of-2 and power-of-2 with memory (SQ(2,3), which uses the same amount of memory as in JBT-d) for nearly the whole regime. Moreover, JBT-d policy achieves nearly the same response time of JIQ when the load is not

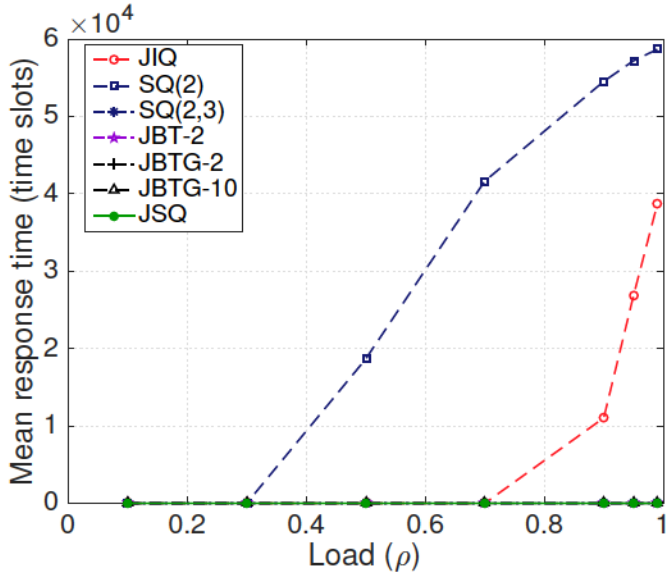

Fig. 3. Throughput performance under 10 heterogeneous servers.

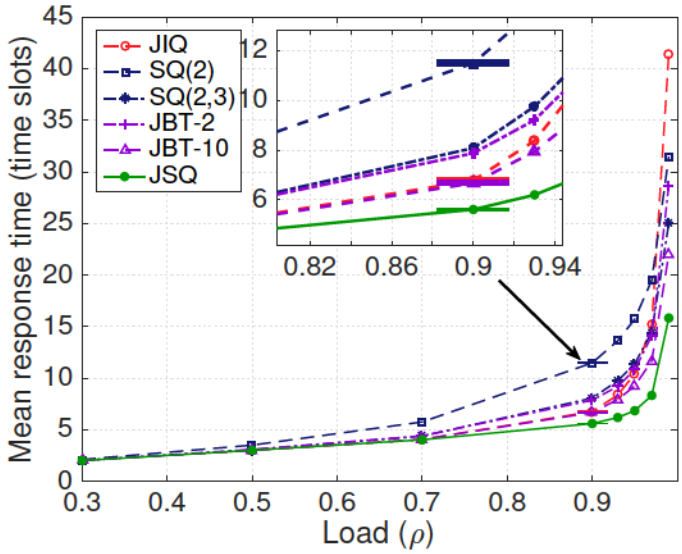

Fig. 4. Delay performance under 10 homogeneous servers.

too high. However, as the load becomes heavier, the performance of JIQ gets worse and worse, and its mean response time is as large as two times of the response time under JBT-d policy when the load is 0.99.

Now, let us get a closer look at the delay performance in heavy-traffic regime, i.e., $\rho > 0.9$, as shown in Figure 5. It can be seen that JBT-10 outperforms JIQ when $\rho > 0.9$ and JBT-2 outperforms

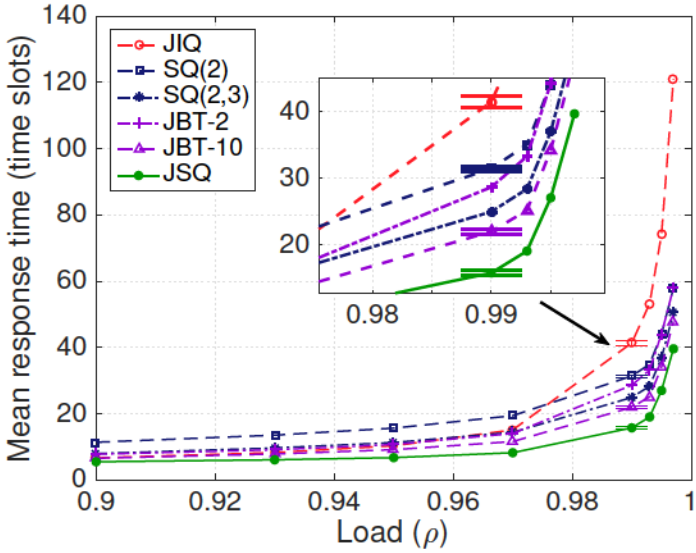

Fig. 5. Heavy-traffic delay performance under 10 homogeneous servers.

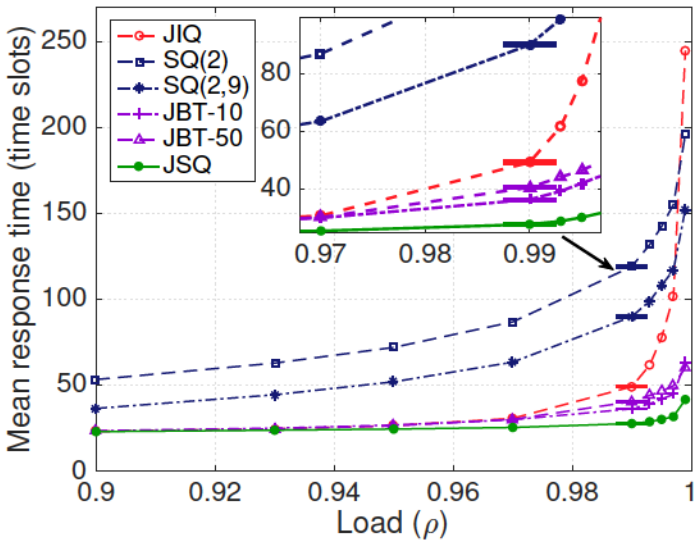

Fig. 6. Heavy-traffic delay performance under 50 homogeneous servers.

JIQ when $\rho > 0.95$ in this case. More importantly, the gap between them keeps increasing as the load gets higher. Note that power-of-2 with memory (SQ(2,3)) also has good performance in this case, which, however, uses a much higher message rate compared to our JBT-d policy, as discussed in the next section.

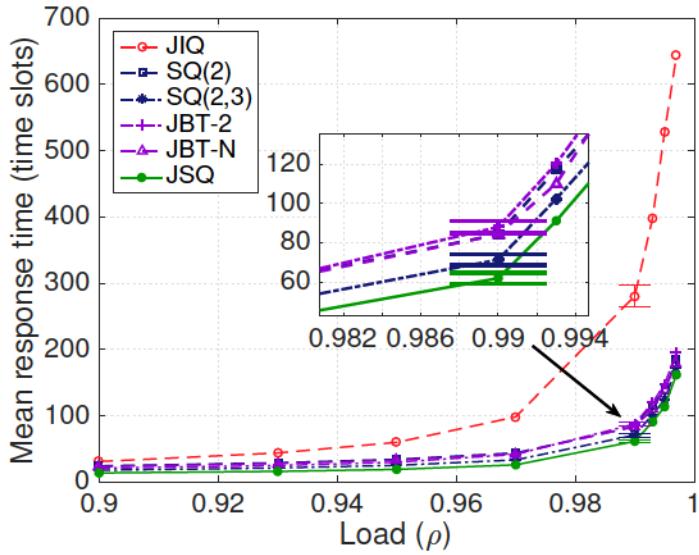

Fig. 7. Heavy-traffic delay performance under 10 homogeneous servers with Poisson arrival and bursty service.

Last, we further provide some results on heavy-traffic delay performance for a larger system size and a bursty service process, respectively. Due to space limitation, the comprehensive results can be found in the technical report [21]. Figure 6 illustrates the heavy-traffic performance under Poisson arrival and Poisson service when $N = 50$. In this case, first thing to note is that even though the power-of-d with memory policy (SQ(2,9)) uses the same amount of memory as in JBT-d, it has a much poorer performance with a much higher message overhead since the message overhead of JBT-d is strictly less than 1 when $T = 1000$ in this case. This means that to improve delay performance in large system size, power-of-d with memory has to increase its message overhead linearly with respect d, while our JBT-d policy is able to achieve good performance with message rate less than 1 even for $d = N$. Moreover, as ρ approaches to 1, the performance of JIQ degrades substantially while our proposed JBT-d remains quite close to JSQ. In Figure 7, the potential number of jobs served in each time slot is either 0 or 10. In this bursty service case, JIQ degrades much faster than that in the Poisson service process. Moreover, in this setting we can easily observe the difference between non-heavy-traffic policy (JIQ) and heavy-traffic optimal policies (all the others). Note that the message overhead of SQ(2,3) is nearly 8 times as large as that of JBT-d, as shown in next section, though its delay is slightly better than JBT-d.

4.3 Message Overhead

We use simulations to further show the low message rate of our proposed JBT-d policy, though a crude upper bound has been established. Here, we consider the 10 homogeneous servers with Poisson arrival and Poisson service, and more results for different settings can be found in the technical report [21]. More specifically, we investigate the impact of different values of T, i.e., the time interval for updating the threshold, on the message rate and its corresponding delay performance at a fixed load $\rho = 0.99$. In particular, we calculate the average number of messages per new job arrival under each policy. For push-based policies, e.g., JSQ, power-of-2 (SQ(2)) and

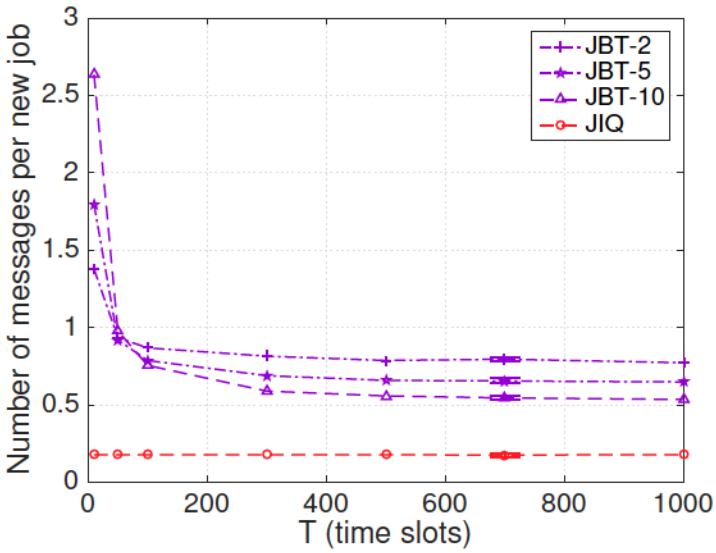

Fig. 8. Message per new job arrival under 10 homogeneous servers with respect to T.

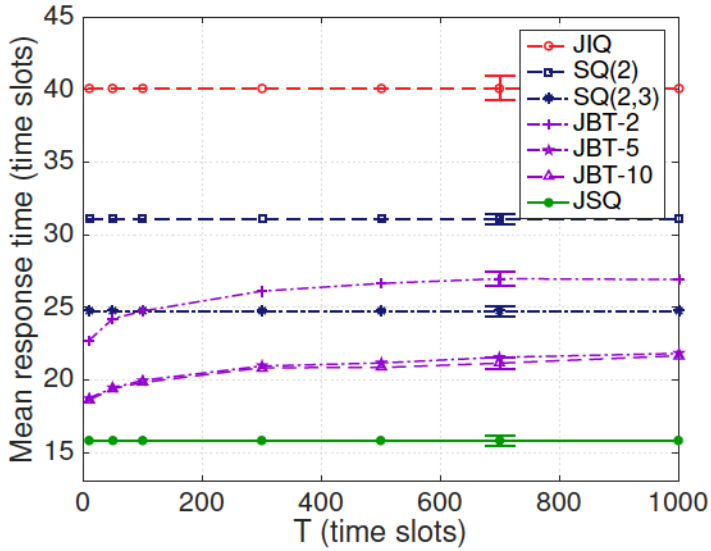

Fig. 9. Delay performance under 10 homogeneous servers with respect to T.

power-of-2 with memory (SQ(2,3)), the message only includes the push-message and is easily calculated as 20, 4, and 4, which is independent of T. For JIQ, we know that the rate is at most one for each new job arrival, which is also independent with T and serves as the benchmark.

Figure 8 shows the message rate of JBT-d with respect to T for different values of d, and the corresponding delay performance is shown in Figure 9. The first thing to note is that the message

rate of JIQ is much smaller than one since the traffic is heavy and hence there are few idle servers in this case, which directly results in the poor performance in the heavy-traffic regime. Second, the message rate of JBT-d is smaller than all push-based policies and becomes less than one when $T > 100$ in this case, which means that it is able to achieve throughput and heavy-traffic delay optimality by requiring a slightly more message than JIQ. Moreover, it can be seen that as T increases, there is no significant change of the delay performance, which indicates that we are allowed to adopt a sufficiently large T while not incurring the loss of performance very much in this case. Last, it is worth noting that a larger d does not necessarily mean a larger message overhead when T is large. This is because when T is large, the push-message in JBT-d will be dominated by the pull-message. For a small d, the number of pull-message may be larger since the threshold may be higher than that under a larger d. As shown in the additional results in the technical report [21], the observations above hold almost for all the considered cases. The exact impact and relationship of T and d would be one of our future research focuses.

5 PROOF OF MAIN RESULTS

The high-level insight for class Π to be heavy-traffic delay optimal is that it always has a preference to shorter queues in the way that is specified by the δ-tilted distribution. The key step behind the proof that JBT-d is heavy-traffic delay optimal is to show that the dispatching distribution for the time slot that is immediately after the threshold update is always a δ-titled distribution.

5.1 Proof of Theorem 3.7

Before we adopt the sufficient conditions in Lemma 3.5 and Lemma 3.6 to prove Theorem 3.7, we first present the following lemmas on the tilted distribution and δ-tilted distribution, respectively.

LEMMA 5.1. *For a system with mean arrival rate $\lambda_\Sigma = \mu_\Sigma - \epsilon$ and a tilted distribution $\mathrm{P}(t)$ under $Z(t)$, we have*

$$\mathbb{E}\left[\langle \mathrm{Q}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t)\right] \le -\epsilon \frac{\mu_{min}}{\mu_\Sigma} \|\mathrm{Q}(t)\| \tag{16}$$

and

$$\mathbb{E}\left[\langle \mathrm{Q}_\perp(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t)\right] \le \epsilon\sqrt{N} \|\mathrm{Q}_\perp(t)\|. \tag{17}$$

PROOF. Consider the left-hand-side (LHS) of Eq. (16)

$$\begin{aligned}
&\mathbb{E}\left[\langle \mathrm{Q}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t)\right] \\
&= \sum_{n=1}^{N} Q_{\sigma_t(n)}(t) \left[\left(\Delta_n(t) + \frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}\right)\lambda_\Sigma - \mu_{\sigma_t(n)}\right] \\
&\overset{(a)}{=} \sum_{n=1}^{N} Q_{\sigma_t(n)}(t)\Delta_n(t)\lambda_\Sigma + \sum_{n}^{N} Q_{\sigma_t(n)}(t)\left(-\epsilon\frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}\right) \\
&\overset{(b)}{\le} \sum_{n=1}^{N} Q_{\sigma_t(n)}(t)\left(-\epsilon\frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}\right) \\
&\overset{(c)}{\le} -\epsilon\frac{\mu_{min}}{\mu_\Sigma}\|\mathrm{Q}(t)\|,
\end{aligned}$$

where equation (a) holds since $\lambda_\Sigma = \mu_\Sigma - \epsilon$; (b) comes from the fact that $\sum_{n=1}^{N} Q_{\sigma_t(n)}(t)\Delta_n(t) \le 0$ under a tilted distribution. This fact is true since $Q_{\sigma_t(1)}(t) \le Q_{\sigma_t(2)}(t) \le \ldots \le Q_{\sigma_t(N)}(t)$ and $\sum_{n=1}^{N}\Delta_n(t) = 0$; inequality (c) follows from the fact that $\|\mathrm{x}\|_1 \ge \|\mathrm{x}\|$ for any $\mathrm{x} \in \mathbb{R}^N$.

Note that $\mathrm{Q}_\perp(t) = \mathrm{Q}(t) - \mathrm{Q}_\parallel(t) = \mathrm{Q}(t) - \frac{\sum Q_n(t)}{N}\mathbf{1} = \mathrm{Q}(t) - Q_{\text{avg}}(t)\mathbf{1}$, in which $Q_{\text{avg}}(t)$ is the average queue length among the N servers at time slot t. Then, consider the left-hand-side (LHS) of Eq. (17)

$$\begin{aligned}
&\mathbb{E}\left[\langle \mathrm{Q}_\perp(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t)\right] \\
=&\sum_{n=1}^{N}\left(Q_{\sigma_t(n)}(t) - Q_{\text{avg}}(t)\right)\left[\left(\Delta_n(t) + \frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}\right)\lambda_\Sigma - \mu_{\sigma_t(n)}\right] \\
\overset{(a)}{=}&\sum_{n=1}^{N} Q_{\sigma_t(n)}(t)\Delta_n(t)\lambda_\Sigma + \sum_{n}^{N}\left(Q_{\sigma_t(n)}(t) - Q_{\text{avg}}(t)\right)\left(-\epsilon\frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}\right) \\
\overset{(b)}{\leq}&\sum_{n=1}^{N}\left(Q_{\sigma_t(n)}(t) - Q_{\text{avg}}(t)\right)\left(-\epsilon\frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}\right) \\
\overset{(c)}{\leq}&\epsilon\sum_{n=1}^{N}\left|\left(Q_{\sigma_t(n)}(t) - Q_{\text{avg}}(t)\right)\right| \\
\overset{(d)}{\leq}&\epsilon\sqrt{N}\left\|\mathrm{Q}_\perp(t)\right\|,
\end{aligned} \tag{18}$$

where equation (a) comes from the facts that $\sum_{n=1}^{N}\Delta_n(t) = 0$ and $\lambda_\Sigma = \mu_\Sigma - \epsilon$; inequality (b) holds since $\sum_{n=1}^{N} Q_{\sigma_t(n)}(t)\Delta_n(t) \leq 0$ under a tilted distribution; inequality (c) is true since $x \leq |x|$ for any $x \in \mathbb{R}$ and $|\epsilon\frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}| \leq \epsilon$ for all $n \in \mathcal{N}$; inequality (d) is true since $\|\mathrm{x}\|_1 \leq \sqrt{N}\|\mathrm{x}\|$ for any $\mathrm{x} \in \mathbb{R}^N$. □

Lemma 5.2. *For a system with mean arrival rate $\lambda_\Sigma = \mu_\Sigma - \epsilon$ and a δ-tilted distribution $\mathrm{P}(t)$ under $Z(t)$, we have*

$$\mathbb{E}\left[\langle \mathrm{Q}_\perp(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t)\right] \leq \sqrt{N}\left\|\mathrm{Q}_\perp(t)\right\|\left(\epsilon - \frac{\delta\lambda_\Sigma}{N}\right). \tag{19}$$

Proof. Consider the left-hand-side (LHS) of Eq. (19), we have

$$\begin{aligned}
&\mathbb{E}\left[\langle \mathrm{Q}_\perp(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t)\right] \\
=&\sum_{n=1}^{N}\left(Q_{\sigma_t(n)}(t) - Q_{\text{avg}}(t)\right)\left[\left(\Delta_n(t) + \frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}\right)\lambda_\Sigma - \mu_{\sigma_t(n)}\right] \\
\overset{(a)}{=}&\sum_{n=1}^{N} Q_{\sigma_t(n)}(t)\Delta_n(t)\lambda_\Sigma + \sum_{n=1}^{N}\left(Q_{\sigma_t(n)}(t) - Q_{\text{avg}}(t)\right)\left(-\epsilon\frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}\right) \\
\overset{(b)}{\leq}&\sum_{n=1}^{N} Q_{\sigma_t(n)}(t)\Delta_n(t)\lambda_\Sigma + \epsilon\sqrt{N}\left\|\mathrm{Q}_\perp(t)\right\| \\
\overset{(c)}{\leq}&-\lambda_\Sigma\delta(Q_{\sigma_t(N)}(t) - Q_{\sigma_t(1)}(t)) + \epsilon\sqrt{N}\left\|\mathrm{Q}_\perp(t)\right\| \\
\overset{(d)}{\leq}&-\lambda_\Sigma\frac{\delta}{\sqrt{N}}\left\|\mathrm{Q}_\perp(t)\right\| + \epsilon\sqrt{N}\left\|\mathrm{Q}_\perp(t)\right\| \\
=&\sqrt{N}\left\|\mathrm{Q}_\perp(t)\right\|(\epsilon - \frac{\delta\lambda_\Sigma}{N}),
\end{aligned}$$

where equation (a) holds since $\sum_{n=1}^{N}\Delta_n(t) = 0$ and $\lambda_\Sigma = \mu_\Sigma - \epsilon$; inequality (b) follows from steps (c) and (d) in Eq. (18); inequality (c) follows from the definition of δ-tilted probability and

the fact that $Q_{\sigma_t(1)}(t) \leq Q_{\sigma_t(2)}(t) \leq \ldots \leq Q_{\sigma_t(N)}(t)$; inequality (d) follows from the fact that $\|\mathrm{Q}_\perp(t)\| \leq \sqrt{N}(Q_{\sigma_t(N)}(t) - Q_{\sigma_t(1)}(t))$. □

Now we are ready to present the proof of Theorem 3.7

Proof of Theorem 3.7: The proof is a direct application of the sufficient conditions for throughput and heavy-traffic delay optimality, i.e., we need only to show Eq. (7) and Eq. (8) hold.

Fix a load balancing policy p in Π. Let us first consider the left-hand-side (LHS) of Eq. (7) with $T_1 = T$,

$$\begin{aligned}
LHS &\overset{(a)}{=} \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\langle \mathrm{Q}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t_0) = Z\right] \\
&\overset{(b)}{=} \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\mathbb{E}\left[\langle \mathrm{Q}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t)\right] | Z(t_0) = Z\right] \\
&\overset{(c)}{\leq} \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[-\epsilon \frac{\mu_{min}}{\mu_\Sigma} \|\mathrm{Q}(t)\| \mid Z(t_0) = Z\right] \\
&\leq -\epsilon \frac{\mu_{min}}{\mu_\Sigma} \|\mathrm{Q}(t_0)\|,
\end{aligned}$$

where equation (a) comes from the linearity of condition expectation; equation (b) follows from the tower property of conditional expectation and the fact that $\mathrm{Q}(t)$, $\mathrm{A}(t)$ and $\mathrm{S}(t)$ are conditionally independent of $Z(t_0)$ when given $Z(t)$. inequality (c) follows from Lemma 5.1 since the policy p adopts a tilted distribution within every time slot for all $Z(t)$. Hence, the condition of Lemma 3.5 is satisfied and thus policy p is throughput optimal.

Let us now turn to consider the left-hand-side (LHS) of Eq. (8) with $T_2 = T$ and $\epsilon < \epsilon_0 = \frac{\delta\mu_\Sigma}{2TN+2\delta}$.

$$\begin{aligned}
LHS &\overset{(a)}{=} \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\langle \mathrm{Q}_\perp(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t_0) = Z\right] \\
&\overset{(b)}{=} \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\mathbb{E}\left[\langle \mathrm{Q}_\perp(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle \mid Z(t)\right] \mid Z(t_0) = Z\right] \\
&\overset{(c)}{\leq} \sum_{t\neq t^*} \mathbb{E}\left[\epsilon\sqrt{N}\|\mathrm{Q}_\perp(t)\| \mid Z(t_0) = Z\right] + \mathbb{E}\left[\sqrt{N}\|\mathrm{Q}_\perp(t^*)\|\left(\epsilon - \frac{\delta\lambda_\Sigma}{N}\right) \mid Z(t_0) = Z\right] \\
&\overset{(d)}{\leq} (T-1)\epsilon\sqrt{N}(\|\mathrm{Q}_\perp(t_0)\| + M) + \left(\epsilon - \frac{\delta\lambda_\Sigma}{N}\right)\sqrt{N}(\|\mathrm{Q}_\perp(t_0)\| - M) \\
&= \left(T\epsilon - \frac{\delta\lambda_\Sigma}{N}\right)\sqrt{N}\|\mathrm{Q}_\perp(t_0)\| + \sqrt{N}M\left(\frac{\delta\lambda_\Sigma}{N} + (T-2)\epsilon\right) \\
&\overset{(e)}{\leq} \left(T\epsilon - \frac{\delta\lambda_\Sigma}{N}\right)\sqrt{N}\|\mathrm{Q}_\perp(t_0)\| + K_2 \\
&\overset{(f)}{\leq} -\frac{\delta\mu_\Sigma}{2N}\sqrt{N}\|\mathrm{Q}_\perp(t_0)\| + K_2,
\end{aligned}$$

where equation (a) comes from the linearity of condition expectation; equation (b) follows from the tower property of conditional expectation and the fact that $\mathrm{Q}_\perp(t)$, $\mathrm{A}(t)$ and $\mathrm{S}(t)$ are conditionally independent of $Z(t_0)$ when given $Z(t)$; inequality (c) follows from Lemmas 5.1 and 5.2 since under policy $p \in \Pi$ there exists at least one time slot t^* within which at least a δ-tilted distribution (or

one of its equivalent distribution in inner-product is δ-tilted) is adopted and for all the other time slots a tilted distribution is used; inequality (d) follows from the fact $|\,\|Q_\perp(t_0+T)\| - \|Q_\perp(t_0)\|\,| \le M = 2T\sqrt{N}\max\{A_{\max}, S_{\max}\}$ and the fact $\epsilon - \frac{\delta\lambda_\Sigma}{N} < 0$ for all $\epsilon < \epsilon_0$; inequality (e) comes from the fact that $\sqrt{N}M(\frac{\delta\lambda_\Sigma}{N} + (T-2)\epsilon) \le K_2 = \sqrt{N}M(\frac{\delta\mu_\Sigma}{N} + T\mu_\Sigma)$, which is independent of ϵ; inequality (f) holds since $\epsilon < \epsilon_0$ and $\lambda_\Sigma = \mu_\Sigma - \epsilon$. Therefore, since both $-\frac{\delta\mu_\Sigma}{2N}\sqrt{N}$ and K_2 are independent of ϵ, the condition of Lemma 3.6 is satisfied, and hence the policy p is heavy-traffic delay optimal. □

5.2 Proof of Proposition 3.14

Let us first look at assertion 1, i.e., JBT-d is in Π under homogeneous servers. Based on Eq. (15), we can conclude that for any $t \ge 0$, the dispatching distribution is a tilted distribution for all $Z(t)$. We are left to show that at time slot $rT+1$, $r \in \{0,1,2,\ldots\}$, the dispatching distribution is at least a δ-distribution for some positive δ. This is equivalent to finding the maximum value for $P_N(rT+1)$ and the minimum value of $P_1(rT+1)$ for all queue length states. In fact, they are achieved at the same time when $\widetilde{p}_N$ is in its largest value based on Eq. (14), which is repeated as follows.

$$P_n(t) = \sum_{i=n}^{N} \widetilde{p}_i(t)\frac{1}{i}.$$

Then, there are two cases to consider.

(a) At time slot $rT+1$, the probability for the event that there are N IDs in memory is equal to 1, i.e., $\widetilde{p}_N(rT+1) = 1$, if and only if all the servers have the same queue length at the end of time slots rT (i.e., sampling slots for updating the threshold), which are also the queue length state at the beginning of $rT+1$, i.e., $Q(rT+1)$. In this case, it can be easily seen that $P_n(rT+1) = \frac{1}{N}$ for all n, which is not a δ-tilted distribution. However, it is an equivalent distribution in inner-product to $\hat{P}_1(rT+1) = 1$ and $\hat{P}_n = 0$ for $2 \le n \le N$ as all the queue lengths are equal, which is indeed a δ-distribution.

(b) If the queue lengths are not all equal at the end of time slots rT, then the maximum value for $\widetilde{p}_N(rT+1)$ is strictly less than 1 and it is obtained when the queue length in the state that there are $N-1$ servers that have the same queue length, which is strictly larger than the remaining one. In this case, by sampling d servers uniformly at random at the end of times slots rT, the probability for the event that there are N IDs in memory, i.e., $\widetilde{p}_N(rT+1)$ is given by

$$\widetilde{p}_N(rT+1) = 1 - \widetilde{p}_1(rT+1) = 1 - \frac{d}{N}.$$

Therefore, we have $P_1(rT+1) = \frac{d}{N} + \frac{N-d}{N^2}$ and $P_n(rT+1) = \frac{N-d}{N^2}$ for $2 \le n \le N$, which is equivalent in inner product to $\hat{P}_1(rT+1) = \frac{d}{N} + \frac{N-d}{N^2}$, $\hat{P}_2(rT+1) = (N-1)\frac{N-d}{N^2}$ and $\hat{P}_n(rT+1) = 0$ for all $3 \le n \le N$ as the $N-1$ queues have the same queue length. As a result, we have for this state $Z(rT+1)$

$$\hat{\Delta}_1(rT+1) = \frac{d}{N}(1-\frac{1}{N}) \text{ and } \hat{\Delta}_N(rT+1) = -\frac{1}{N}.$$

Thus, it is a δ-distribution with $\delta = \min\{\frac{d}{N}(1-\frac{1}{N}), \frac{1}{N}\}$, which is the lower bound for δ. That is, for any state $Z(rT+1)$, the dispatching distribution is at least a δ-distribution. Therefore, every $T+1$ time slots, there exists one time slot in which the dispatching distribution (or an inner product equivalent distribution) is at least a δ-tilted distribution with $\delta = \min\{\frac{d}{N}(1-\frac{1}{N}), \frac{1}{N}\}$.

The proof for heterogeneous servers follows exact the same idea with additional care on the service rate. The probability for the server $\sigma_t(n)$ to be selected at time t, i.e., $P_n(t)$ is given by

$$P_n(t) = \mu_{\sigma_t(n)} \sum_{i=n}^{N} \frac{\widetilde{p}_i(t)}{\sum_{\{j \in m(t), |m(t)|=i\}} \mu_j}.$$

From it we can easily see that if $\Delta_n(t) = P_n(t) - \frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}$ is positive, then we must have that $\Delta_{n-1}(t)$ is also positive as it has one more term in the equation above. Therefore, we can find a k between 2 and N such that $\Delta_n(t) = P_n(t) - \frac{\mu_{\sigma_t(n)}}{\mu_\Sigma} \geq 0$ for all $n < k$ and $\Delta_n(t) \leq 0$ for all $n \geq k$. Therefore, condition (i) of Π is satisfied.

For the condition (ii), we need to find the maximum value of $\widetilde{p}_N(rT+1)$ to bound δ. There are also two cases as before.

(a) If $\widetilde{p}_N(rT+1) = 1$, then we must have that the queue lengths are all equal at the end of time slots rT, which is the same as that at the beginning of time slot $rT+1$. In this case, $P_n(rT+1) = \frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}$ for all n. Note that this dispatching distribution is an equivalent distribution in inner-product to $\hat{P}_1(rT+1) = 1$ and $\hat{P}_n = 0$ for $2 \leq n \leq N$ as all the queue lengths are equal, which is a δ-distribution.

(b) If $\widetilde{p}_N(rT+1) \neq 1$, the maximum value of $\widetilde{p}_N(rT+1)$ is obtained when there are $N-1$ servers that have the same queue length, which is strictly larger than the remaining one. In this case, we have $\widetilde{p}_N(rT+1) = 1 - \widetilde{p}_1(rT+1) = 1 - \frac{d}{N}$ as before. Thus, we can obtain

$$P_1(rT+1) = \frac{d}{N} + \left(1 - \frac{d}{N}\right) \frac{\mu_{\sigma_t(1)}}{\mu_\Sigma},$$

and $P_n(rT+1) = (1 - \frac{d}{N}) \frac{\mu_{\sigma_t(n)}}{\mu_\Sigma}$ for $2 \leq n \leq N$. This is equivalent in inner product to $\hat{P}_1(rT+1) = P_1(rT+1)$, $\hat{P}_2(rT+1) = \sum_{n=2}^{N} P_n(rT+1)$ and $\hat{P}_n(rT+1) = 0$ for all $3 \leq n \leq N$ since the last $N-1$ servers have the same queue lengths. As a result, we have for this $Z(rT+1)$

$$\hat{\Delta}_1(rT+1) = \frac{d}{N}\left(1 - \frac{\mu_{\sigma_t(1)}}{\mu_\Sigma}\right) \text{ and } \hat{\Delta}_N(rT+1) = -\frac{\mu_{\sigma_t(N)}}{\mu_\Sigma}.$$

Thus, it is a δ-distribution with $\delta = \min\{\frac{d}{N}(1 - \frac{\mu_{max}}{\mu_\Sigma}), \frac{\mu_{min}}{\mu_\Sigma}\}$, in which $\mu_{max} = \max_{n \in \mathcal{N}} \mu_n$ and $\mu_{min} = \min_{n \in \mathcal{N}} \mu_n$, which is the lower bound of δ. Hence, for any $Z(rT+1)$, the dispatching probability distribution (or its inner product equivalent one) is at least a δ-distribution. □

6 CONCLUSION

We introduce a class Π of flexible load balancing policies, which are shown to be throughput and heavy-traffic delay optimal. This class includes as special cases JSQ, power-of-d, and also allows flexibility in designing other new policies. The JIQ policy, albeit exhibiting a good performance when the traffic load is not heavy, is not in Π since it is not heavy-traffic delay optimal even for homogeneous servers. A new policy called JBT-d is proposed in the class Π, which enjoys the simplicity of JIQ while guaranteeing heavy-traffic delay optimal. A unified analytic framework is established to characterize this class of policies by exploring their common characteristics and provide sufficient conditions that guarantee the heavy-traffic delay optimality. Extensive simulations are used to demonstrate the good performance and low complexity of the proposed policy compared to other existing ones.

REFERENCES

[1] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1 (2008), 107–113.

[2] Atilla Eryilmaz and R Srikant. 2012. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Systems* 72, 3-4 (2012), 311–359.
[3] G Foschini and JACK Salz. 1978. A basic dynamic routing problem and diffusion. *IEEE Transactions on Communications* 26, 3 (1978), 320–327.
[4] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. 2008. Cloud computing and grid computing 360-degree compared. In *2008 Grid Computing Environments Workshop*. Ieee, 1–10.
[5] Lars George. 2011. *HBase: the definitive guide.* " O'Reilly Media, Inc.".
[6] Varun Gupta, Mor Harchol Balter, Karl Sigman, and Ward Whitt. 2007. Analysis of join-the-shortest-queue routing for web server farms. *Performance Evaluation* 64, 9 (2007), 1062–1081.
[7] Bruce Hajek. 1982. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability* (1982), 502–525.
[8] Yi Lu, Qiaomin Xie, Gabriel Kliot, Alan Geller, James R Larus, and Albert Greenberg. 2011. Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services. *Performance Evaluation* 68, 11 (2011), 1056–1071.
[9] Siva Theja Maguluri, R Srikant, and Lei Ying. 2014. Heavy traffic optimal resource allocation algorithms for cloud computing clusters. *Performance Evaluation* 81 (2014), 20–39.
[10] Michael Mitzenmacher. 2001. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems* 12, 10 (2001), 1094–1104.
[11] Michael Mitzenmacher, Balaji Prabhakar, and Devavrat Shah. 2002. Load balancing with memory. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*. IEEE, 799–808.
[12] Debankur Mukherjee, Sem C Borst, Johan SH Van Leeuwaarden, Philip A Whiting, et al. 2016. Universality of load balancing schemes on the diffusion scale. *Journal of Applied Probability* 53, 4 (2016), 1111–1124.
[13] Rajesh Nishtala, Hans Fugal, Steven Grimm, Marc Kwiatkowski, Herman Lee, Harry C Li, Ryan McElroy, Mike Paleczny, Daniel Peek, Paul Saab, et al. 2013. Scaling memcache at facebook. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. 385–398.
[14] Devavrat Shah and Balaji Prabhakar. 2002. The use of memory in randomized load balancing. In *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*. IEEE, 125.
[15] Alexander L Stolyar. 2015. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Systems* 80, 4 (2015), 341–361.
[16] John N Tsitsiklis and Kuang Xu. 2013. Queueing system topologies with limited flexibility. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 41. ACM, 167–178.
[17] Nikita Dmitrievna Vvedenskaya, Roland L'vovich Dobrushin, and Fridrikh Izrailevich Karpelevich. 1996. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problemy Peredachi Informatsii* 32, 1 (1996), 20–34.
[18] Weina Wang, Kai Zhu, Lei Ying, Jian Tan, and Li Zhang. 2016. MapTask scheduling in MapReduce with data locality: Throughput and heavy-traffic optimality. *IEEE/ACM Transactions on Networking* 24, 1 (2016), 190–203.
[19] Richard R Weber. 1978. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability* (1978), 406–413.
[20] Lei Ying, R Srikant, and Xiaohan Kang. 2015. The power of slightly more than one sample in randomized load balancing. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 1131–1139.
[21] Xingyu Zhou, Fei Wu, Jian Tan, Yin Sun, and Ness Shroff. 2017. Designing Low-Complexity Heavy-Traffic Delay-Optimal Load Balancing Schemes: Theory to Algorithms. http://arxiv.org/abs/1710.04357. (Oct. 2017).

A PROOF OF LEMMA 3.5

Before we present the proof of Lemma 3.5, we first introduce two lemmas which will be the key ingredients in the proof. The first lemma enables us to bound the moments of a stationary distribution based on drift condition, which can be simplified by the second lemma.

The following lemma is introduced in [18], which is an extension of Lemma 1 in [2] and can be proved from the results in [7].

LEMMA A.1. *For an irreducible aperiodic and positive recurrent Markov chain $\{X(t), t \geq 0\}$ over a countable state space $\mathcal{X}$, which converges in distribution to $\overline{X}$, and suppose $V : \mathcal{X} \to \mathbb{R}_+$ is a Lyapunov function. We define the T time slot drift of V at X as*

$$\Delta V(X) := [V(X(t_0 + T)) - V(X(t_0))]\mathcal{I}(X(t_0) = X),$$

where $\mathcal{I}(.)$ is the indicator function. Suppose for some positive finite integer T, the T time slot drift of V satisfies the following conditions:

- *(C1) There exists an $\gamma > 0$ and a $\kappa < \infty$ such that for any $t_0 = 1, 2, \ldots$ and for all $X \in \mathcal{X}$ with $V(X) \geq \kappa$,*

$$\mathbb{E}\left[\Delta V(X) \mid X(t_0) = X\right] \leq -\gamma.$$

- *(C2) There exists a constant $D < \infty$ such that for all $X \in \mathcal{X}$,*

$$\mathbb{P}(|\Delta V(X)| \leq D) = 1.$$

Then $\{V(X(t)), t \geq 0\}$ converges in distribution to a random variable $\overline{V}$, and there exists constants $\theta^* > 0$ and $C^* < \infty$ such that $\mathbb{E}\left[e^{\theta^* \overline{V}}\right] \leq C^*$, which directly implies that all moments of random variable $\overline{V}$ exist and are finite. More specifically, there exist finite constants $\{M_r, r \in \mathbb{N}\}$ such that for each positive r, $\mathbb{E}\left[V(\overline{X})^r\right] \leq M_r$, where M_r are fully determined by κ, γ and D.

Lemma A.2. *For any $t \geq 0$, we have*

$$\|\mathrm{Q}(t+1)\|^2 - \|\mathrm{Q}(t)\|^2 \leq 2\langle \mathrm{Q}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle + K \tag{20}$$

where K is a finite constant.

Proof. Consider the left-hand-side (LHS) of Eq. (20).

$$\begin{aligned} LHS &= \|\mathrm{Q}(t) + \mathrm{A}(t) - \mathrm{S}(t) + \mathrm{U}(t)\|^2 - \|\mathrm{Q}(t)\|^2 \\ &\overset{(a)}{\leq} \|\mathrm{Q}(t) + \mathrm{A}(t) - \mathrm{S}(t)\|^2 - \|\mathrm{Q}(t)\|^2 \\ &= 2\langle \mathrm{Q}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle + \|\mathrm{A}(t) - \mathrm{S}(t)\|^2 \\ &\overset{(b)}{\leq} 2\langle \mathrm{Q}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle + K \end{aligned}$$

where inequality (a) holds as $[\max(a, 0)]^2 \leq a^2$ for any $a \in \mathbb{R}$; in inequality (b), $K \triangleq N \max(A_{\max}, S_{\max})^2$ holds due to the assumptions that $A_\Sigma(t) \leq A_{\max}$ and $S_n(t) \leq S_{\max}$ for all $t \geq 0$ and all $n \in \mathcal{N}$, and independent of the queue length. □

We are now ready to prove Lemma 3.5.

Proof of Lemma 3.5: The proof follows from the application of Lemma A.1 to the Markov chain $\{Z^{(\epsilon)}(t), t \geq 0\}$ with Lyapunov function $V(Z^{(\epsilon)}) := \left\|\mathrm{Q}^{(\epsilon)}\right\|$ and $T = T_1$ since $m^{(\epsilon)}(t)$ is always finite. In particular, this proof is completed in two steps, where the superscript $^{(\epsilon)}$ will be omitted for ease of notations.

(i) First, in order to apply Lemma A.1, we need to show that the Markov chain $\{Z(t), t \geq 0\}$ is irreducible, aperiodic and positive recurrent under the hypothesis of Lemma 3.5. It can be easily seen that $\{Z(t), t \geq 0\}$ is irreducible and aperiodic. Thus, we are left with the task to prove that the Markov chain is positive recurrent. By the extension of Foster-Lyapunov theorem, it suffices to find a Lyapunov function and a positive constant T such that the expected T time slot Lyapunov drift is bounded within a finite subset of the state space and negative outside this subset.

Consider the Lyapunov function $W(Z) := \|\mathrm{Q}\|^2$, and the corresponding expected T_1 time slot mean conditional Lyapunov drift under the hypothesis of Lemma 3.5.

$$\begin{aligned}
&\mathbb{E}\left[W(Z(t_0+T_1)) - W(Z(t_0)) \mid Z(t_0)\right] \\
=&\mathbb{E}\left[\|\mathrm{Q}(t_0+T_1)\|^2 - \|\mathrm{Q}(t_0)\|^2 \mid Z(t_0)\right] \\
=&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T_1-1} \left(\|\mathrm{Q}(t+1)\|^2 - \|\mathrm{Q}(t)\|^2\right) \mid Z(t_0)\right] \\
\overset{(a)}{\leq}&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T_1-1} 2\langle \mathrm{Q}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle + K \mid Z(t_0)\right] \\
\overset{(b)}{\leq}&-2\gamma\|\mathrm{Q}(t_0)\| + 2K_1 + KT_1
\end{aligned} \tag{21}$$

where inequality (a) follows from Lemma A.2, and inequality (b) results directly from the hypothesis in Eq. (7). Pick any $\beta > 0$ and let $\mathcal{B} = \{Z \in \mathcal{S} : \|\mathrm{Q}\| \leq \frac{2K_1+KT_1+\beta}{2\gamma}\}$. Then $\mathcal{B}$ is a finite subset of $\mathcal{S}$ as $m(t)$ is finite. Moreover, for any $Z \in \mathcal{B}$, the conditional mean drift is less or equal to $2K_1 + KT_1$, and for any $Z \in \mathcal{B}^c$, it is less than or equal to $-\beta$. This finishes the proof of positive recurrence for any $\epsilon > 0$, and hence throughput optimal.

(ii) Second, in order to show that the hypothesis in Lemma 3.5 also ensures the bounded moments for the stationary distribution, we will resort to Lemma A.1. Thus, we need to check Conditions (C1) and (C2), respectively.

For Condition (C1), we have

$$\begin{aligned}
&\mathbb{E}\left[\Delta V(Z) \mid Z(t_0) = Z\right] \\
=&\mathbb{E}\left[\|\mathrm{Q}(t_0+T_1)\| - \|\mathrm{Q}(t_0)\| \mid Z(t_0) = Z\right] \\
=&\mathbb{E}\left[\sqrt{\|\mathrm{Q}(t_0+T_1)\|^2} - \sqrt{\|\mathrm{Q}(t_0)\|^2} \mid Z(t_0) = Z\right] \\
\overset{(a)}{\leq}&\frac{1}{2\|\mathrm{Q}(t_0)\|}\mathbb{E}\left[\|\mathrm{Q}(t_0+T_1)\|^2 - \|\mathrm{Q}(t_0)\|^2 \mid Z(t_0) = Z\right] \\
\overset{(b)}{\leq}&-\gamma + \frac{2K_1 + KT_1}{2\|\mathrm{Q}(t_0)\|}
\end{aligned}$$

where inequality (a) follows from the fact that $f(x) = \sqrt{x}$ is concave; (b) comes from the upper bound in Eq. (21). Hence, (C1) in Lemma A.1 is verified.

For Condition (C2), we have

$$\begin{aligned}
|\Delta V(Z)| =& \left|\|\mathrm{Q}(t_0+T_1)\| - \|\mathrm{Q}(t_0)\|\right| \mathcal{I}(Z(t_0) = Z) \\
\overset{(a)}{\leq}& \|\mathrm{Q}(t_0+T_1) - \mathrm{Q}(t_0)\| \mathcal{I}(Z(t_0) = Z) \\
\overset{(b)}{\leq}& T_1\sqrt{N}\max(A_{\max}, S_{\max})
\end{aligned}$$

where inequality (a) follows from the fact that $|\,\|\mathrm{x}\| - \|\mathrm{y}\|\,| \leq \|\mathrm{x} - \mathrm{y}\|$ holds for any $\mathrm{x}, \mathrm{y} \in \mathbb{R}^N$; inequality (b) holds due to the assumptions that the $A_\Sigma(t) \leq A_{\max}$ and $S_n(t) \leq S_{\max}$ for all $t \geq 0$ and all $n \in \mathcal{N}$, and independent of the queue length. This verifies Condition (C2) and hence complete the proof of Lemma 3.5. □

B PROOF OF LEMMA 3.6

We now proceed to prove Lemma 3.6. Before we present the proof, the following lemmas which serve as useful preliminary steps are first introduced. Denote by $\mathrm{Q}_{\parallel}$ and $\mathrm{Q}_{\perp}$ the parallel and perpendicular components of the queue length vector Q with respect to the line $\mathrm{c} = \frac{1}{\sqrt{N}}\mathbf{1}$, i.e.,

$$\mathrm{Q}_{\parallel} := \langle \mathrm{c}, \mathrm{Q}\rangle \mathrm{c} \qquad \mathrm{Q}_{\perp} := \mathrm{Q} - \mathrm{Q}_{\parallel} \tag{22}$$

The following lemma is a natural extension of Lemma 7 in [2] to T time slots.

Lemma B.1. *Define the following Lyapunov functions*

$$V_{\perp}(Z) := \|\mathrm{Q}_{\perp}\|, W(Z) := \|\mathrm{Q}\|^2 \text{ and } W_{\parallel}(Z) := \|\mathrm{Q}_{\parallel}\|^2$$

with the corresponding T time-slot drift given by

$$\begin{aligned}
\Delta V_{\perp}(Z) &:= [V_{\perp}(Z(t_0+T)) - V_{\perp}(Z(t_0))]\mathcal{I}(Z(t_0) = Z)\\
\Delta W(Z) &:= [W(Z(t_0+T)) - W(Z(t_0))]\mathcal{I}(Z(t_0) = Z)\\
\Delta W_{\parallel}(Z) &:= [W_{\parallel}(Z(t_0+T)) - W_{\parallel}(Z(t_0))]\mathcal{I}(Z(t_0) = Z)
\end{aligned}$$

Then, the drift of $V_{\perp}(.)$ can be bounded in terms of $W(.)$ and $W_{\parallel}(.)$ as follows.

$$\Delta V_{\perp}(Z) \le \frac{1}{2\|\mathrm{Q}_{\perp}\|}(\Delta W(Z) - \Delta W_{\parallel}(Z))$$

for all $Z \in \mathcal{S}$.

Lemma B.2. *For any $t \ge 0$, we have*

$$\|\mathrm{Q}_{\parallel}(t+1)\|^2 - \|\mathrm{Q}_{\parallel}(t)\|^2 \ge 2\langle \mathrm{Q}_{\parallel}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle.$$

Proof.

$$\begin{aligned}
&\|\mathrm{Q}_{\parallel}(t+1)\|^2 - \|\mathrm{Q}_{\parallel}(t)\|^2\\
=&2\langle \mathrm{Q}_{\parallel}(t), \mathrm{Q}_{\parallel}(t+1) - \mathrm{Q}_{\parallel}(t)\rangle + \|\mathrm{Q}_{\parallel}(t+1) - \mathrm{Q}_{\parallel}(t)\|^2\\
\ge&2\langle \mathrm{Q}_{\parallel}(t), \mathrm{Q}_{\parallel}(t+1) - \mathrm{Q}_{\parallel}(t)\rangle\\
=&2\langle \mathrm{Q}_{\parallel}(t), \mathrm{Q}(t+1) - \mathrm{Q}(t)\rangle - 2\langle \mathrm{Q}_{\parallel}(t), \mathrm{Q}_{\perp}(t+1) - \mathrm{Q}_{\perp}(t)\rangle\\
\overset{(a)}{\ge}&2\langle \mathrm{Q}_{\parallel}(t), \mathrm{Q}(t+1) - \mathrm{Q}(t)\rangle\\
\overset{(b)}{\ge}&2\langle \mathrm{Q}_{\parallel}(t), \mathrm{A}(t) - \mathrm{S}(t)\rangle
\end{aligned}$$

where the inequality (a) is true because $\langle \mathrm{Q}_{\parallel}(t), \mathrm{Q}_{\perp}(t)\rangle = 0$ and $\langle \mathrm{Q}_{\perp}(t+1), \mathrm{Q}_{\parallel}(t)\rangle = 0$; (b) follows from the fact that all the components of $\mathrm{Q}_{\parallel}(t)$ and $\mathrm{U}(t)$ are nonnegative. □

We are now ready to prove the following result, which is often called *state space collapse* and is the key ingredient for establishing heavy traffic delay optimality. It shows that under the hypothesis of Lemma 3.6, the multi-dimension space for the queue length vector will reduce to one dimension in the sense that the deviation from the line c is bounded by a constant, which is independent with the heavy-traffic parameter ϵ.

Lemma B.3. *If the assumptions in Lemma 3.6 hold, then we have that $\mathrm{Q}_{\perp}$ is bounded in the sense that in steady state there exists finite constants $\{L_r, r \in \mathbb{N}\}$ such that*

$$\mathbb{E}\left[\left\|\overline{\mathrm{Q}}_{\perp}^{(\epsilon)}\right\|^r\right] \le L_r$$

for all $\epsilon \in (0, \epsilon_0)$ and $r \in \mathbb{N}$.

Proof. It suffices to show that $V_\perp(Z)$ satisfies the Conditions (C1) and (C2) in Lemma A.1. Fix $\epsilon \in (0, \epsilon_0)$, and the superscript will be omitted for simplicity in the following arguments.

(i) For the Condition (C1), let $\Lambda(t) := \|Q(t+1)\|^2 - \|Q(t)\|^2$ and $\Lambda_\parallel(t) := \|Q_\parallel(t+1)\|^2 - \|Q_\parallel(t)\|^2$. Then, we have

$$\begin{aligned}
&\mathbb{E}\left[\Delta V_\perp(Z) \mid Z(t_0) = Z\right] \\
\overset{(a)}{\leq} &\frac{1}{2\|Q_\perp\|}\mathbb{E}\left[\Delta W(Z) - \Delta W_\parallel(Z) \mid Z(t_0) = Z\right] \\
= &\frac{1}{2\|Q_\perp\|}\mathbb{E}\left[\sum_{t=t_0}^{t_0+T_2-1} \Lambda(t) - \Lambda_\parallel(t) \mid Z(t_0) = Z\right] \\
\overset{(b)}{\leq} &\frac{1}{2\|Q_\perp(t_0)\|}\mathbb{E}\left[\sum_{t=t_0}^{t_0+T_2-1} 2\langle Q_\perp(t), A(t) - S(t)\rangle + K \mid Z(t_0) = Z\right] \\
\overset{(c)}{\leq} &-\eta + \frac{2K_2 + KT_2}{2\|Q_\perp(t_0)\|},
\end{aligned}$$

where the inequality (a) follows from Lemma B.1; the inequality (b) holds as a result of Lemmas A.2 and B.2; the inequality (c) follows directly from the assumption in Eq. (8). Hence, the Condition (C1) is verified.

(ii) For the Condition (C2), we have

$$\begin{aligned}
&|\Delta V_\perp(Z)| \\
= &|\,\|Q_\perp(t_0+T_2)\| - \|Q_\perp(t_0)\|\,|\mathcal{I}(Z(t_0) = Z) \\
\overset{(a)}{\leq} &\|Q_\perp(t_0+T_2) - Q_\perp(t_0)\|\,\mathcal{I}(Z(t_0) = Z) \\
= &\|Q(t_0+T_2) - Q_\parallel(t_0+T_2) - Q(t_0) + Q_\parallel(t_0)\|\,\mathcal{I}(Z(t_0) = Z) \\
\overset{(b)}{\leq} &\|Q(t_0+T_2) - Q(t_0)\| + \|Q_\parallel(t_0+T_2) - Q_\parallel(t_0)\|\,\mathcal{I}(Z(t_0) = Z) \\
\overset{(c)}{\leq} &2\|Q(t_0+T_2) - Q(t_0)\|\,\mathcal{I}(Z(t_0) = Z) \\
\overset{(d)}{\leq} &2T_2\sqrt{N}\max(A_{\max}, S_{\max})
\end{aligned} \tag{23}$$

where the inequality (a) follows from the fact that $|\,\|x\| - \|y\|\,| \leq \|x - y\|$ holds for any $x, y \in \mathbb{R}^N$; inequality (b) follows from triangle inequality; (c) holds due to the non-expansive property of projection to a convex set. (d) holds due to the assumptions that the $A_\Sigma(t) \leq A_{\max}$ and $S_n(t) \leq S_{\max}$ for all $t \geq 0$ and all $n \in \mathcal{N}$, and independent of the queue length. This verifies Condition (C2) and hence complete the proof of Lemma B.3. □

The following result on the unused service is another key ingredient for establishing heavy-traffic delay optimal.

Lemma B.4. *For any $\epsilon > 0$ and $t \geq 0$, we have*

$$Q_n^{(\epsilon)}(t+1)U_n^{(\epsilon)}(t) = 0 \text{ and } q^{(\epsilon)}(t+1)u^{(\epsilon)}(t) = 0.$$

If the system has a finite first moment, then we have for some constants c_1 and c_2

$$\mathbb{E}\left[\left\|\overline{U}^{(\epsilon)}\right\|^2\right] \leq c_1\epsilon \text{ and } \mathbb{E}\left[(\overline{u}^{(\epsilon)})^2\right] \leq c_2\epsilon$$

Proof. According to the queue dynamic in Eq. (1), we can see when $U_n(t)$ is positive, $Q_n(t+1)$ must be zero, which gives the results $Q_n^{(\epsilon)}(t+1)U_n^{(\epsilon)}(t) = 0$ for all $n \in \mathcal{N}$ and all $t \geq 0$, and the corresponding result for the resource-pooled system $q^{(\epsilon)}(t+1)u^{(\epsilon)}(t) = 0$.

Then, let us consider the Lyapunov function $W_1(Z(t)) = \|\mathrm{Q}(t)\|_1$. In the steady state with a finite first moment, the mean drift of $W_1(Z(t))$ is zero. Then, we have

$$0 = \mathbb{E}\left[\left\|\mathrm{A}^{(\epsilon)}\right\|_1 - \|\mathrm{S}\|_1 + \left\|\overline{\mathrm{U}}^{(\epsilon)}\right\|_1\right]$$

which directly implies

$$\mathbb{E}\left[\left\|\overline{\mathrm{U}}^{(\epsilon)}\right\|_1\right] = \epsilon \tag{24}$$

Moreover, due to the fact that $U_n(t) \leq S_{\max}$ for all $n \in \mathcal{N}$ and $t \geq 0$, we have $\left\|\overline{\mathrm{U}}^{(\epsilon)}\right\|^2 \leq S_{\max}\left\|\overline{\mathrm{U}}^{(\epsilon)}\right\|_1$. Therefore, we can conclude that $\mathbb{E}\left[\left\|\overline{\mathrm{U}}^{(\epsilon)}\right\|^2\right] \leq S_{\max}\epsilon$ and $\mathbb{E}\left[\left(\overline{u}^{(\epsilon)}\right)^2\right] \leq NS_{\max}\epsilon$. □

Now, we are well prepared to prove Lemma 3.6

Proof of Lemma 3.6: First, let us consider the Lyapunov function $V_1(Z) := \|\mathrm{Q}\|_1^2$ and the corresponding conditional mean drift, defined as $D_1(Z(t)) := \mathbb{E}\left[V_1(Z(t+1)) - V_1(Z(t)) \mid Z(t) = Z\right]$.

Then, we have the following equation, in which the time reference (t) will be omitted after the second step for brevity and $\mathrm{Q}^+ := \mathrm{Q}(t+1)$.

$$\begin{aligned}
&D_1(Z(t))\\
=&\mathbb{E}\left[\|\mathrm{Q}(t+1)\|_1^2 - \|\mathrm{Q}(t)\|_1^2 \mid Z(t) = Z\right]\\
=&\mathbb{E}\left[\left(\|\mathrm{Q}(t)\|_1 + \|\mathrm{A}(t)\|_1 - \|\mathrm{S}(t)\|_1 + \|\mathrm{U}(t)\|_1\right)^2 \mid Z(t) = Z\right] - \mathbb{E}\left[\|\mathrm{Q}(t)\|_1^2 \mid Z(t) = Z\right]\\
=&\mathbb{E}\left[2\|\mathrm{Q}\|_1\left(\|\mathrm{A}\|_1 - \|\mathrm{S}\|_1\right) + \left(\|\mathrm{A}\|_1 - \|\mathrm{S}\|_1\right)^2 + 2\left(\|\mathrm{Q}\|_1 + \|\mathrm{A}\|_1 - \|\mathrm{S}\|_1\right)\|\mathrm{U}\|_1 + \|\mathrm{U}\|_1^2 \mid Z\right]\\
=&\mathbb{E}\left[2\|\mathrm{Q}\|_1\left(\|\mathrm{A}\|_1 - \|\mathrm{S}\|_1\right) + \left(\|\mathrm{A}\|_1 - \|\mathrm{S}\|_1\right)^2 + 2\|\mathrm{Q}^+\|_1\|\mathrm{U}\|_1 - \|\mathrm{U}\|_1^2 \mid Z\right]\\
\leq&\mathbb{E}\left[2\|\mathrm{Q}\|_1\left(\|\mathrm{A}\|_1 - \|\mathrm{S}\|_1\right) + \left(\|\mathrm{A}\|_1 - \|\mathrm{S}\|_1\right)^2 + 2\|\mathrm{Q}^+\|_1\|\mathrm{U}\|_1 \mid Z\right]
\end{aligned} \tag{25}$$

Under the assumptions of Lemma 3.6, there exists a steady-state distribution with finite moments for any $\epsilon > 0$. Therefore, the mean drift in steady-state is zero, i.e., $\mathbb{E}\left[D_1(\overline{Z}^{(\epsilon)})\right] = 0$. Therefore, taking the expectation of both sides of Eq. (25) with respect to the steady-state distribution $\overline{Z}^{(\epsilon)}$, yields

$$\epsilon\mathbb{E}\left[\sum_{n=1}^{N}\overline{Q}_n^{(\epsilon)}\right] \leq \frac{\zeta^{(\epsilon)}}{2} + \mathbb{E}\left[\left\|\overline{\mathrm{Q}}^{(\epsilon)}(t+1)\right\|_1\left\|\overline{\mathrm{U}}^{(\epsilon)}(t)\right\|_1\right]$$

where $\zeta^{(\epsilon)} = (\sigma_\Sigma^{(\epsilon)})^2 + \nu_\Sigma^2 + \epsilon^2$. For the resource-pooled system, by letting $N = 1$ in Eq. (25) and taking the expectation with respect to $\overline{q}^{(\epsilon)}$, we have

$$\epsilon\mathbb{E}\left[\overline{q}^{(\epsilon)}\right] = \frac{\zeta^{(\epsilon)}}{2} + \mathbb{E}\left[\overline{q}^{(\epsilon)}(t+1)u^{(\epsilon)}(t)\right] - \frac{1}{2}\mathbb{E}\left[(u^{(\epsilon)})^2\right].$$

Then, based on the property on the unused service in Lemma B.4, we have

$$\frac{\zeta^{(\epsilon)}}{2} - \frac{c_2}{2}\epsilon \leq \epsilon\mathbb{E}\left[\overline{q}^{(\epsilon)}\right] \leq \epsilon\mathbb{E}\left[\sum_{n=1}^{N}\overline{Q}_n^{(\epsilon)}\right] \leq \frac{\zeta^{(\epsilon)}}{2} + \overline{B}^{(\epsilon)} \tag{26}$$

where $\overline{B}^{(\epsilon)} := \mathbb{E}\left[\left\|\overline{Q}^{(\epsilon)}(t+1)\right\|_1 \left\|\overline{U}^{(\epsilon)}(t)\right\|_1\right]$.

Therefore, in order to show heavy-traffic delay optimality, all we need to show is that $\lim_{\epsilon\downarrow 0}\overline{B}^{(\epsilon)} = 0$. Note that $\overline{B}^{(\epsilon)}$ can be bounded as follows.

$$\begin{aligned}\overline{B}^{(\epsilon)} &\overset{(a)}{=} N\mathbb{E}\left[\langle \overline{U}^{(\epsilon)}(t), -\overline{Q}_{\perp}^{(\epsilon)}(t+1)\rangle\right]\\ &\overset{(b)}{\leq} N\sqrt{\mathbb{E}\left[\left\|\overline{U}_{\perp}^{(\epsilon)}(t)\right\|^2\right]\mathbb{E}\left[\left\|\overline{Q}_{\perp}^{(\epsilon)}(t+1)\right\|^2\right]}\\ &\overset{(c)}{=} N\sqrt{\mathbb{E}\left[\left\|\overline{U}_{\perp}^{(\epsilon)}(t)\right\|^2\right]\mathbb{E}\left[\left\|\overline{Q}_{\perp}^{(\epsilon)}(t)\right\|^2\right]},\end{aligned}$$

where the equality (a) comes from the property $Q_n^{(\epsilon)}(t+1)U_n^{(\epsilon)}(t) = 0$ for all $n \in \mathcal{N}$ and all $t \geq 0$ in Lemma B.4 and the definition of $Q_{\perp}$; the inequality (b) holds due to Cauchy-Schwartz inequality; the last equality (c) is true since the distributions of $\overline{Q}_{\perp}^{(\epsilon)}(t+1)$ and $\overline{Q}_{\perp}^{(\epsilon)}(t)$ are the same in steady state.

As shown in Lemma B.3, $\mathbb{E}\left[\left\|\overline{Q}_{\perp}^{(\epsilon)}\right\|^2\right] \leq L_2$ holds for all $\epsilon \in (0, \epsilon_0)$ and some constant L_2 which is independent of ϵ. Meanwhile, note that $\mathbb{E}\left[\left\|\overline{U}^{(\epsilon)}\right\|^2\right] \leq c_1\epsilon$ for some c_1 independent of ϵ based on Lemma B.4. Then, we have for all $\epsilon \in (0, \epsilon_0)$

$$\overline{B}^{(\epsilon)} \leq N\sqrt{c_1\epsilon L_2} \tag{27}$$

Therefore, it can be seen from Eq. (27) that $\lim_{\epsilon\downarrow 0}\overline{B}^{(\epsilon)} = 0$, which directly implies $\lim_{\epsilon\downarrow 0}\epsilon\mathbb{E}\left[\sum_n \overline{Q}_n^{(\epsilon)}\right] = \lim_{\epsilon\downarrow 0}\epsilon\mathbb{E}\left[\overline{q}^{(\epsilon)}\right]$, and thus the proof of Lemma 3.6 is completed. □