Performance Evaluation 145 (2021) 102146

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Asymptotically optimal load balancing in large-scale heterogeneous systems with multiple dispatchers[☆]

Xingyu Zhou[a,*], Ness Shroff[b], Adam Wierman[c]

[a] Department of ECE, The Ohio State University, Columbus, USA
[b] Department of ECE and CSE, The Ohio State University, Columbus, USA
[c] Department of Computing and Mathematical Sciences, Caltech, Pasadena, USA

ARTICLE INFO

Article history:
Available online 8 October 2020

Keywords:
Asymptotically optimal
Load balancing
Heterogeneous systems
Multiple dispatchers
Delayed information

ABSTRACT

We consider the load balancing problem in large-scale heterogeneous systems with multiple dispatchers. We introduce a general framework called Local-Estimation-Driven (LED). Under this framework, each dispatcher keeps local (possibly outdated) estimates of the queue lengths for all the servers, and the dispatching decision is made purely based on these local estimates. The local estimates are updated via infrequent communications between dispatchers and servers. We derive sufficient conditions for LED policies to achieve throughput optimality and delay optimality in heavy-traffic, respectively. These conditions directly imply delay optimality for many previous local-memory based policies in heavy traffic. Moreover, the results enable us to design new delay optimal policies for heterogeneous systems with multiple dispatchers. Finally, the heavy-traffic delay optimality of the LED framework also sheds light on a recent open question on how to design optimal load balancing schemes using delayed information.

1. Introduction

Load balancing, which is responsible for dispatching jobs on parallel servers, has attracted significant interest in recent years. This is motivated by the challenges associated with efficiently dispatching jobs in large-scale data centers and cloud applications, which are rapidly increasing in size. A good load balancing policy not only ensures high throughput by maximizing server utilization, but also improves the user experience by minimizing delay.

There have been numerous load balancing policies proposed in the literature. The most straightforward one is Join-Shortest-Queue (JSQ), which has been shown to enjoy optimal delay in both non-asymptotic (for homogeneous servers) and asymptotic regimes [1–3]. However, it is difficult to implement in today's large-scale data centers due to the large message overhead between the dispatcher and servers. As a result, alternative load balancing policies with low message overhead have been proposed. For example, the Power-of-d policy [4] has been shown to achieve optimal average delay in heavy traffic with only $2d$ messages per arrival [5]. Another common load balancing policy is the pull-based Join-Idle-Queue (JIQ) [6,7], which has been shown to outperform the Power-of-d policy using less overhead. However, both Power-of-d and JIQ mainly achieve good performance for systems with homogeneous servers. Recently, some works consider heterogeneous servers and propose flexible and low message overhead policies that achieve optimal delay in

[☆] This project has been funded in part through NSF, USA grants: CNS-2007231, CNS-1719371, and CNS-1717060 and NSF, USA grants AitF-1637598 and CNS-1518941.

[*] Corresponding author.

E-mail addresses: zhou.2055@osu.edu (X. Zhou), shroff.11@osu.edu (N. Shroff), adamw@caltech.edu (A. Wierman).

https://doi.org/10.1016/j.peva.2020.102146
0166-5316/© 2020 Elsevier B.V. All rights reserved.

heavy traffic [8,9]. However, only a single dispatcher is considered in these works. Theoretical analysis of load balancing with multiple dispatchers has mainly focused on the JIQ policy so far [10,11], which has a poor performance in heavy traffic and is even generally unstable for heterogeneous systems [8].

Note that heterogeneous systems with multiple dispatchers are now almost the default scenarios in today's cloud infrastructures. On one hand, the heterogeneity comes from the usage of multiple generations of CPUs and various types of devices [12]. On the other hand, with the massive amount of data, a scalable cloud infrastructure needs multiple dispatchers to increase both throughput and robustness [13].

Motivated by this, a recent work [14] proposes a new framework named Local Shortest Queue (LSQ) for designing load balancing policies for heterogeneous systems with multiple dispatchers. In particular, under this framework, each dispatcher keeps its own, local, and possibly outdated view of each server's queue length. Upon arrival, each dispatcher routes to the server with shortest local view. A small amount of message overhead is used to update the local view. The authors successfully establish sufficient conditions on the update scheme for the system to be stable. Moreover, extensive simulations were conducted to show that LSQ policies significantly outperform well-known low-communication policies while using similar communication overhead in both heterogeneous and homogeneous cases. However, no theoretical guarantee on the delay performance is provided and the authors mention it as an important future research direction. It is worth noting that the key challenge for establishing a delay performance guarantee for this framework is that it only uses possibly outdated local information to dispatch jobs. In fact, the problem of designing delay optimal load balancing schemes that only have access to delayed information has recently been listed as an open problem in [15].

Inspired by this, in this paper, we are particularly interested in the following questions: *Is it possible to establish delay performance guarantees for load balancing in heterogeneous systems with multiple dispatchers? If so, can these guarantees be achieved using only delayed information?*

Contributions. To answer the questions above, we propose a general framework of load balancing for heterogeneous systems with multiple dispatchers that uses only delayed (out-of-date) information about the system state. We call this framework Local-Estimation-Driven (LED) and it generalizes the LSQ framework. Our main results provide sufficient conditions for LED policies to be both throughput optimal and delay optimal in heavy-traffic. Our key contributions can be summarized as follows.

First, we introduce the LED framework for designing load balancing policies for heterogeneous systems with multiple dispatchers. In this framework, each dispatcher keeps its own local estimates of queue lengths for all the servers, and makes its dispatching decision based purely on its own local estimates according to a certain dispatching strategy. The local estimates are updated infrequently via an update strategy that is based on communications between dispatchers and servers.

Second, we derive sufficient conditions for LED policies to be throughput optimal and delay optimal in heavy-traffic. The importance of the sufficient conditions is three-fold: (i) It can be shown that previous local-memory based policies (e.g., LSQ) satisfy our sufficient conditions. As a result, we are able to show that they are not only throughput optimal (in a stronger sense) but also delay optimal in heavy-traffic. (ii) The conditions allow us to design new delay optimal load balancing policies with zero dispatching delay and low message overhead that work for heterogeneous servers and multiple dispatchers. (iii) These conditions also provide us with a systematic approach for generalizing previous optimal policies to the case of multiple dispatchers and exploring the trade-off between memory (i.e., local estimations) and message overhead. For instance, we are able to show that the Power-of-d policy can achieve delay optimality in heavy traffic, even in heterogeneous systems, as long as the imbalance among the service rates is not too large.

Third, the LED framework also sheds light on the open problem posed in [15], which asks how to design heavy-traffic delay optimal policies that only use delayed information. Our main results for LED policies not only demonstrate that it is possible to achieve optimal delay in heavy-traffic via only delayed information, but also highlight conditions on the extent to which old information is useful. Moreover, they provide methods for using the delayed information to achieve optimality in heavy traffic. Interestingly, the LED framework also shows that, in the case of multiple dispatchers, inaccurate information can actually lead to improved performance.

To establish the main results, we need to address the following two technical challenges. First, each dispatcher in our model only has access to delayed and outdated system information. Second, we consider a large class of dispatching strategies specified by a general condition. To handle the general condition, we have to apply a refined drift analysis to obtain the necessary negative drifts required for throughput optimality and delay optimality. In order to handle the outdated queue length information, we have to transfer the drift on local estimates to the corresponding drift on the actual queue lengths. To this end, we develop a new Lyapunov function, and combine this with sample-path analysis, and couplings arguments to obtain tight bounds.

Related work. The study of efficient load balancing algorithms has been a hot topic for a long time and spans across different asymptotic regimes. The most extensively investigated policy might be Join-Shortest-Queue (JSQ), under which the incoming jobs are always sent to the server with the shortest queue length. JSQ has been shown to be optimal in a stochastic order sense [1,16] and also heavy-traffic delay optimal [2,3]. To overcome the high-complexity of JSQ, several low-complexity schemes have been proposed, including Power-of-d [4] and Join-Idle-Queue (JIQ) [6,7]. Recently, different classes of policies are proposed in [8,9], which are able to obtain both advantages of Power-of-d (e.g., heavy-traffic delay optimality) and JIQ (e.g., low-message overhead and zero dispatching time).

Compared to the large literature on the single dispatcher case, there are relatively few works that consider multiple dispatchers, and they mainly focus on the JIQ policy. In particular, [10] presents a new large-system asymptotic analysis of JIQ without the simplifying assumptions in [6]. The property of asymptotically zero waiting time of JIQ was generalized to the case of multiple dispatchers in [11]. However, the results for JIQ in [6,10,11] all assume that the loads at various dispatchers are strictly equal. Without this assumption, [17] shows that the waiting time under JIQ no longer vanishes in the large-system regime and two enhanced JIQ schemes are proposed. As mentioned earlier, although JIQ is a scalable choice for the multiple-dispatcher case, it is not delay optimal in heavy traffic for homogeneous servers and not even generally stable for heterogeneous systems [8].

The case of heterogeneous systems with multiple dispatchers has received very little attention from the theoretical community so far. To the best of our knowledge, the recent framework proposed in [14] is the first attempt to study efficient load balancing schemes with a theoretical guarantee for the scenario of heterogeneous systems with multiple dispatchers. In particular, under the proposed Local-Shortest-Queue (LSQ) framework, each dispatcher independently keeps its own local view of server queue lengths and routes jobs to the shortest among them. Communication is used only to update the local views and make sure that they are not too far from the real queue lengths. The main contributions of [14] are the sufficient conditions for any LSQ policy to achieve strong stability with low message overhead. Additionally, extensive simulations have been used to demonstrate its appeal. Nevertheless, theoretical guarantees on the delay of LSQ policies remain an important unsolved question.

It is worth pointing out that the idea of using local memory to hold possibly old information for load balancing was also explored in two recent works [18,19]. As we discuss later, these two proposed policies are in our LED framework. Both works only consider a single dispatcher and homogeneous servers, which is also a special case of our model. Further, their analysis focuses on the large-system asymptotic regime where the number of servers goes to infinity, while our analysis deals with a finite number of servers.

2. System model and preliminaries

This section describes the system model and assumptions considered in this paper. Then, several necessary preliminaries are presented.

2.1. System model

We consider a discrete-time (i.e., time-slotted) load balancing system consisting of M dispatchers and N possibly-heterogeneous servers. Each server maintains an infinite capacity FIFO queue. At each dispatcher, there is a local memory, through which the dispatcher can have some (possibly delayed) information about the system states. In each time-slot, the central dispatcher routes the new incoming tasks to one of the servers, immediately upon arrival. Once a task joins a queue, it remains in that queue until its service is completed. Each server is assumed to be work conserving, i.e., a server is idle if and only if its corresponding queue is empty.

2.1.1. Arrivals

Let $A^m(t)$ denote the number of exogenous tasks that arrive at dispatcher m at the beginning of time-slot t. We assume that $A_\Sigma(t) = \sum_{m=1}^M A^m(t)$ is an integer-valued random variable, which is *i.i.d.* across time-slots. The mean and variance of $A_\Sigma(t)$ are denoted by λ_Σ and σ_Σ^2, respectively. We further assume that there is a positive probability that $A_\Sigma(t)$ is zero. The allocation of total arriving tasks among the M dispatchers is allowed to use any arbitrary policy that is independent of system states. Note that, in contrast to previous works on multiple dispatchers [6,10,11], we do not require that the loads at all dispatchers are equal. We assume that there is a strictly positive probability for tasks to arrive at each dispatcher at any time-slot t. That is, there exists a strictly positive constant p_0 such that

$$\mathbb{P}\left(A^m(t) > 0\right) \geq p_0, \quad \forall (m,t) \in \mathcal{M} \times \mathbb{N}, \tag{1}$$

where $\mathcal{M} = \{1, 2, \ldots, M\}$. Moreover, we assume that $A^m(t)$ is *i.i.d.* across time-slots with mean arrival rate denoted by λ_m. We further let $A_n^m(t)$ denote the number of new arrivals at server n from dispatcher m at the beginning of time-slot t. Let $A_n(t) = \sum_{m=1}^M A_n^m(t)$ be the total number of arriving tasks at server n at the beginning of time-slot t.

2.1.2. Service

Let $S_n(t)$ denote the amount of service that server n offers for queue n in time-slot t. That is, $S_n(t)$ is the maximum number of tasks that can be completed by server n at time-slot t. We assume that $S_n(t)$ is an integer-valued random variable, which is *i.i.d.* across time-slots. We also assume that $S_n(t)$ is independent across different servers as well as the arrival process. The mean and variance of $S_n(t)$ are denoted as μ_n and ν_n^2, respectively. Let $\mu_\Sigma \triangleq \Sigma_{n=1}^N \mu_n$ and $\nu_\Sigma^2 \triangleq \Sigma_{n=1}^N \nu_n^2$ denote the mean and variance of the hypothetical total service process $S_\Sigma(t) \triangleq \sum_{n=1}^N S_n(t)$. Let $\epsilon = \mu_\Sigma - \lambda_\Sigma$ characterize the distance between the arrival rate and the boundary of capacity region.

2.1.3. Queue dynamics

Let $Q_n(t)$ be the queue length of server n at the beginning of time slot t. Let $A_n(t)$ denote the number of tasks routed to queue n at the beginning of time-slot t according to the dispatching decision. Then the evolution of the length of queue n is given by

$$Q_n(t+1) = Q_n(t) + A_n(t) - S_n(t) + U_n(t), n = 1, 2, \dots, N, \tag{2}$$

where $U_n(t) = \max\{S_n(t) - Q_n(t) - A_n(t), 0\}$ is the unused service due to an empty queue.

We do not assume any specific distribution for arrival and service processes. Moreover, in contrast to previous works [3,8], we do not require that both arrival and service processes have a finite support. Instead, we only need the condition that their distributions are light-tailed. More specifically, we assume that

$$\mathbb{E}\left[e^{\theta_1 A_\Sigma(t)}\right] \le D_1 \text{ and } \mathbb{E}\left[e^{\theta_2 S_n(t)}\right] \le D_2, \tag{3}$$

for each n where the constants $\theta_1 > 0$, $\theta_2 > 0$, $D_1 < \infty$ and $D_2 < \infty$ are all independent of ϵ.

2.2. Local-estimation-driven (LED) framework

We are interested in the case that the local memory at each dispatcher m stores an estimate of the queue length for each server n. In particular, we let $\widetilde{Q}_n^m(t)$ be the local estimate of the queue length for server n from dispatcher m at the beginning of time-slot t (before any arrivals and departures). More specifically, we introduce the following framework for load balancing.

Definition 1. A Local-Estimation-Driven (LED) policy is composed of the following components:

(a) **Dispatching strategy:** At the beginning of each time-slot, each dispatcher m chooses one of the servers for new arrivals *purely* based on its local estimates (i.e., local queue length estimates $\widetilde{\mathbf{Q}}^m$)
(b) **Update strategy:** At the end of each time-slot, each dispatcher would possibly update its local estimates, e.g., synchronize local queue length estimate with the true queue length.

The definition of LED is broad, and it includes a variety of classical load balancing policies. For example, it can be seen to include LSQ policy studied in [14], by choosing the dispatching strategy to be that new arrivals at each dispatcher are dispatched to the queue with the shortest local estimate. Moreover, it also includes two recent local memory based policies in [18,19] that are developed for the case of single dispatcher and homogeneous servers.

To study LED, we model the system as a discrete-time Markov chain $\{Z(t) = (\mathbf{Q}(t), m(t)), t \ge 0\}$ with state space $\mathcal{Z}$, using the queue length vector $\mathbf{Q}(t)$ together with the memory state $m(t) \triangleq (\widetilde{\mathbf{Q}}^1(t), \widetilde{\mathbf{Q}}^2(t), \dots, \widetilde{\mathbf{Q}}^m(t))$. We consider a set of load balancing systems $\{Z^{(\epsilon)}(t), t \ge 0\}$ parameterized by ϵ such that the mean arrival rate of the total exogenous arrival process $\{A_\Sigma^{(\epsilon)}(t), t \ge 0\}$ is $\lambda_\Sigma^{(\epsilon)} = \mu_\Sigma - \epsilon$. Note that the parameter ϵ characterizes the distance between the arrival rate and the boundary of the capacity region. We are interested in the throughput performance and the steady-state delay performance in the heavy-traffic regime under any LED policy.

A load balancing system is stable if the Markov chain $\{Z(t), t \ge 0\}$ is positive recurrent, and $\overline{Z} = \{\overline{\mathbf{Q}}, \overline{m}\}$ denotes the random vector whose distribution is the same as the steady-state distribution of $\{Z(t), t \ge 0\}$. We have the following definition.

Definition 2 (*Throughput Optimality*)**.** A load balancing policy is said to be throughput optimal if for any arrival rate within the capacity region, i.e., for any $\epsilon > 0$, the system is positive recurrent and all the moments of $\left\|\overline{\mathbf{Q}}^{(\epsilon)}\right\|$ are finite.

Note that this is a stronger definition of throughput optimality than that in [14,20,21] because, besides the positive recurrence, it also requires all the moments to be finite in steady state for any arrival rate within the capacity region.

To characterize the steady-state average delay performance in the heavy-traffic regime when ϵ approaches zero, by Little's law, it is sufficient to focus on the summation of all the queue lengths. First, recall the following fundamental lower bound on the expected sum queue lengths in a load balancing system under any throughput optimal policy [3]. Note that this result was originally proved with the assumption of finite support on the service process (Lemma 5 in [3]), which can be generalized to service processes with light-tailed distributions with a careful analysis of the unused service, see our proof of Lemma 6.

Lemma 1. *Given any throughput optimal policy and assuming that $(\sigma_\Sigma^{(\epsilon)})^2$ converges to a constant σ_Σ^2 as ϵ decreases to zero, then*

$$\liminf_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\sum_{n=1}^{N} \overline{Q}_n^{(\epsilon)}\right] \ge \frac{\zeta}{2}, \tag{4}$$

where $\zeta \triangleq \sigma_\Sigma^2 + \nu_\Sigma^2$.

The right-hand-side of Eq. (4) is the heavy-traffic limit of a hypothesized single-server system with arrival process $A_{\Sigma}^{(\epsilon)}(t)$ and service process $\sum_{n}^{N} S_n(t)$ for all $t \geq 0$. This hypothetical single-server queueing system is often called the *resource-pooled system*. Since a task cannot be moved from one queue to another in the load balancing system, it is easy to see that the expected sum queue lengths of the load balancing system is larger than the expected queue length in the resource-pooled system. However, if a policy achieves the lower bound in Eq. (4) in the heavy-traffic limit, based on Little's law this policy achieves the minimum average delay of the system in steady-state, and is thus said to be heavy-traffic delay optimal, see [3,5,8,20–22].

Definition 3 (*Heavy-traffic Delay Optimality in Steady-state*)**.** A load balancing scheme is said to be heavy-traffic delay optimal in steady-state if the steady-state queue length vector $\overline{\mathbf{Q}}^{(\epsilon)}$ satisfies

$$\limsup_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\sum_{n=1}^{N} \overline{Q}_n^{(\epsilon)}\right] \leq \frac{\zeta}{2},$$

where ζ is defined in Lemma 1.

2.3. Dispatching preference

In order to provide a unified way to specify the dispatching strategy in LED, we first introduce a concept called *dispatching preference*. In particular, let $P_n^m(t)$ be the probability that new arrivals at dispatcher m are dispatched to server n at time-slot t. We define $\beta_n^m(t) \triangleq P_n^m(t) - \frac{\mu_n}{\mu_\Sigma}$, which is the difference in probability that server n will be chosen under a particular dispatching strategy and random routing (weighted by service rate). Then, we have the following definition.

Definition 4 (*Dispatching Preference*)**.** Fix a dispatcher m, let $\sigma_t(\cdot)$ be a permutation of $(1, 2, \ldots, N)$ that satisfies

$$\widetilde{Q}_{\sigma_t(1)}^m(t) \leq \widetilde{Q}_{\sigma_t(2)}^m(t) \leq \ldots \leq \widetilde{Q}_{\sigma_t(N)}^m(t).$$

The dispatching preference at dispatcher m is a N-dimensional vector denoted by $\Delta^m(t)$, the nth component of which is given by $\Delta_n^m(t) \triangleq \beta_{\sigma_t(n)}^m(t)$.

In words, the dispatching preference at a dispatcher m specifies how servers with different local estimates are preferred in a unified way such that it is independent of the actual values of local estimates. It only depends on the relative order of local estimates. More specifically, fix a dispatcher m, by definition we can see that weighted random routing strategy has no preference for any servers and $\Delta_n^m(t) = 0$ for any n. On the other hand, if new arrivals are always dispatched to the server with the shortest local estimate (e.g, LSQ policy), we have $\Delta_1^m(t) > 0$ and $\Delta_n^m(t) < 0$ for all $2 \leq n \leq N$. Thus, we can see that a positive value for $\Delta_n^m(t)$ means that the dispatching strategy has a preference for the server with the nth shortest local estimation.

Definition 5 (*δ-tilted Sum Condition*)**.** Fix a dispatcher m, for all $1 \leq j \leq N-1$, $\sum_{n=1}^{j} \Delta_n^m(t) \geq \delta$ for some constant $\delta \geq 0$ at each time-slot t.

Note that a similar concept of dispatching preference was introduced in [8], which defines a class of load balancing policies for the case of a single dispatcher with up-to-date information. In comparison, the δ-tilted sum condition introduced above is more general, even in the same scenario (i.e., $M = 1$ and local memory has up-to-date information). In fact, it can be easily shown that any policy in the class introduced in [8] satisfies the δ-tilted sum condition, However, there are various policies that satisfy the δ-tilted sum condition are not within the class in [8]. As shown in the next section, one important by-product of the δ-tilted sum condition is that it can be used to show that Power-of-d can also achieve throughput optimality and heavy-traffic delay optimality even in heterogeneous servers as long as the imbalance among the service rates satisfies a certain condition. This result cannot be obtained by using the result in [8].

3. Main results

In this section, we first present the sufficient conditions for LED policies to be throughput optimal and heavy-traffic delay optimal. Then, we explore several example policies within LED framework to demonstrate its flexibility in designing new load balancing schemes.

3.1. Sufficient conditions

Let us begin with the sufficient conditions for LED policies to be throughput optimal. In particular, we specify conditions for the dispatching strategy and update strategy that guarantee throughput optimality.

To state the theorem, we need the following notation. Let $\mathcal{I}_n^m(t)$ be an indicator function which equals 1 if and only if the local estimate of server n's queue length at dispatcher m gets updated, i.e., the estimated queue length $\widetilde{Q}_n^m(t)$ is set to the actual queue length $Q_n(t)$ at the end of time-slot t.

Theorem 1. *Consider an LED policy. Suppose the dispatching strategy satisfies the δ-tilted sum condition for some $\delta \geq 0$ and the update strategy can guarantee the condition that there exists a positive constant p such that*

$$\mathbb{E}\left[\mathcal{I}_n^m(t) \mid Z(t) = Z\right] \geq p \tag{5}$$

holds for all Z and $(m, n, t) \in \mathcal{M} \times \mathcal{N} \times \mathbb{N}$. Then, this policy is throughput optimal, i.e., the system under this policy is positive recurrent with all the moments being bounded for any $\epsilon > 0$.

Proof. See Section 5.1 □

Note that this theorem directly implies that LSQ is not only strongly stable but also enables the system to have all the moments bounded in steady-state. Moreover, it suggests that any dispatching strategy that is as good as (weighted) random routing is sufficient to guarantee throughput optimality. Further, the update probability can be a function of the traffic load.

Now, we turn to presenting the sufficient conditions for LED policies to be delay optimal in heavy traffic. In order to achieve delay optimality, we need stronger conditions on both the dispatching strategy and the update strategy.

Theorem 2. *Consider an LED policy. Suppose the dispatching strategy at each dispatcher satisfies the δ-tilted sum condition with a uniform lower bound $\delta > 0$. Suppose the update strategy can guarantee that there exists a positive constant p such that*

$$\mathbb{E}\left[\mathcal{I}_n^m(t) \mid Z(t) = Z\right] \geq p \tag{6}$$

holds for all Z and $(m, n, t) \in \mathcal{M} \times \mathcal{N} \times \mathbb{N}$, independent of past updates. Moreover, both δ and p are independent of ϵ. Then, this policy is heavy-traffic delay optimal.

Proof. See Section 5.2 □

This theorem not only establishes a delay performance guarantee for many previous local-memory based policies (e.g., LSQ in [14], low-message policies in [18,19]), but also provides us with the flexibility to design new delay optimal load balancing for different scenarios with heterogeneous servers and multiple dispatchers, as discussed in the next section. More importantly, our results directly suggest that it is possible to use only delayed information to achieve delay optimality, which resolves one of the open problems listed in [15].

Challenges and high-level proof idea. The key challenges to obtaining our main results are: (i) outdated queue length information at each dispatcher and (ii) a large class of dispatching strategies specified by δ-tilted sum condition (which is the most general condition to the best of our knowledge). For (ii), we need a refined analysis of the drift towards the origin (which is required for throughput optimality) and the drift towards the line where all queue lengths are equal (which is required for delay-optimality in heavy traffic). Loosely speaking, we successfully show that for any $\delta \geq 0$, the δ-tilted sum condition can guarantee a drift towards the origin. For any $\delta > 0$ (independent of ϵ), it can guarantee a drift towards the line $\mathbf{1} = (1, 1, \ldots, 1)$. However, all the drifts are with respect to local estimates since each dispatcher is only aware of these local estimates rather than the true queue lengths. Thus, we need to further transfer the drifts on local estimates to the actual queueing systems based on the update strategy. To this end, let us consider two queueing systems: a local-estimation system at each dispatcher and the actual system (i.e., queue lengths at servers). For throughput optimality, the drift towards the origin on local estimates implies that each local-estimation system will not blow up. Meanwhile, the update strategy guarantees that the local-estimation system is not far away from the actual queueing system in expectation. Therefore, the expected sum queue lengths of the actual queueing system will also not blow up. Although the idea is intuitive, there are still some technical challenges. One is that we need to design a new Lyapunov function which includes local estimates as well to apply Foster–Lyapunov theorem, since they can also be unbounded. For delay optimality in heavy traffic, it is not obvious (as throughput optimality) that we can easily transfer the drift based on the implication that the local-estimation is not far away from the actual queueing system in expectation. To handle this, we need a careful sample path analysis and coupling of the two systems to obtain tight bounds. Roughly speaking, we show that for each time-slot there is always a positive probability that shorter queues in the actual systems are preferred (hence a drift towards the line $\mathbf{1} = (1, 1, \ldots, 1)$). In addition, we can upper bound the error that occurs during the transferring process due to outdated queue lengths information. Combining the two parts, yields the delay optimality result in heavy-traffic.

3.2. Examples

To illustrate the applications of Theorems 1 and 2, in this section, we introduce examples of LED policies that are both throughput optimal and heavy-traffic delay optimal. The flexibility provided by our sufficient conditions not only allows us to include previous policies as special cases, but also enables us to design new flexible policies.

3.2.1. Dispatching strategy

Let us first introduce some typical dispatching strategies that satisfy δ-tilted sum condition with $\delta > 0$.

Example 1 (*Local–Join-Shortest-Queue (L-JSQ)*)**.** At the beginning of each time-slot t, the dispatcher forwards its arrivals to the server with the shortest local estimate with ties broken arbitrarily. That is, consider dispatcher m, the chosen server is $i^* \in \arg\min_n \{\widetilde{Q}_n^m\}$.

This dispatching strategy is the same as that in the LSQ policy in [14]. By the definition of dispatching preference, we can see that under L-JSQ, $\Delta_1^m(t) = 1 - \mu_{\sigma_t(1)}/\mu_\Sigma > 0$ and $\Delta_n^m(t) = -\mu_{\sigma_t(n)}/\mu_\Sigma < 0$. Hence, it satisfies δ-tilted sum condition even for heterogeneous servers with $\delta = \mu_{min}/\mu_\Sigma$ where $\mu_{min} = \min_n \mu_n$.

Instead of always joining the server with the shortest local estimate, it is also possible to join a server whose queue length is below a threshold while satisfying the δ-tilted sum condition.

Example 2 (*Local–Join-Below-Average (L-JBA)*)**.** At the beginning of each time-slot t, the dispatcher forwards its arrivals to a randomly chosen server whose local estimate is below or equal to the average local queue length estimation. That is, consider dispatcher m with the average local estimate being $\bar{Q}^m(t) = \frac{1}{N}\sum_n \widetilde{Q}_n^m(t)$. Let $\mathcal{A} \triangleq \{n : \widetilde{Q}_n^m(t) \le \bar{Q}^m(t)\}$. Then, for each $i \in \mathcal{A}$, $P_i^m(t) = \mu_i / \sum_{n\in\mathcal{A}} \mu_n$, and for $i \notin \mathcal{A}$, $P_i^m(t) = 0$.

It can be easily shown from the definition that L-JBA also satisfies δ-tilted sum condition. Note that, compared to L-JSQ, in the heterogeneous case, it needs the dispatcher to know the service rate of each server, which can be easily obtained by the update strategies introduced next. This strategy is more flexible than L-JSQ since it does not require new arrivals to be only sent to the server with the shortest local estimate, which could be used in the scenarios with data locality. Moreover, some randomness in the dispatching strategy is also useful, as discussed in the next section.

Further, it is possible to generalize many previous heavy-traffic delay optimal policies into the LED framework. For example, we can directly apply the Power-of-d policy as our dispatching strategy.

Example 3 (*Local–Power-of-d (L-Pod)*)**.** At the beginning of each time-slot t, the dispatcher randomly chooses $d \ge 2$ servers and sends arrivals to the server that has the shortest local estimation among the d servers.

It can be easily shown that L-Pod satisfies the δ-tilted sum condition with $\delta = \frac{1}{N}$ for homogeneous servers. However, for heterogeneous servers, Power-of-d is not stable in general [7] and hence L-Pod is not either. In the following, inspired by [23], we show that as long as the service rate imbalance among servers satisfies a certain condition, L-Pod would also satisfy the δ-tilted sum condition.

Proposition 1. *Suppose the service rate vector $\boldsymbol{\mu} \in \mathbb{R}_+^N$ satisfies*

$$\frac{\sum_{n=1}^{j} \mu_{[n]}}{\mu_\Sigma} + \delta \le 1 - \frac{\binom{N-j}{d}}{\binom{N}{d}} \qquad \forall 1 \le j \le N-1, \tag{7}$$

for some constant $\delta \ge 0$, in which $\mu_{[n]}$ is the nth largest service rate. Then, L-Pod satisfies the δ-tilted sum condition.

Proof. See Appendix D. □

It can be seen from Proposition 1 that the condition on the imbalance of service rates depends on the value of d. If $d = 1$ (i.e., Power-of-d reduces to random routing), then the only possible values of $\boldsymbol{\mu}$ and δ that satisfy Eq. (7) are $\mu_n = \mu$ for all n and $\delta = 0$. On the other extreme case when $d = N$, then all $\boldsymbol{\mu} \in \mathbb{R}_+^N$ satisfy Eq. (7) with $\delta = \frac{\mu_{[N]}}{\mu_\Sigma} > 0$.

3.2.2. Update strategy

Now, let us turn to discussing update strategies that satisfy the condition in Theorem 2. In particular, the update strategy can either be push-based (dispatcher samples servers) or pull-based (servers report to dispatchers).

Definition 6 (*Push-Update*)**.** If there are new arrivals, then at the end of the time-slot the dispatcher m samples d distinct servers with a positive probability $\hat{p}$. Then, it updates the corresponding d local estimates with the true values.

It has been shown in [14] that even for $d = 1$, the push-update strategy is guaranteed to satisfy the condition in Theorem 2.

Definition 7 (*Pull-Update*)**.** At the end of each time-slot, for each server n if there are completed tasks, then the server will uniformly at random pick a dispatcher m and then abide by one of the following two rules:

- If the server becomes idle (i.e., no tasks), it sends $(n, 0)$ to dispatcher m.
- If not, it sends (n, Q_n) to dispatcher m with probability $\hat{p}$.

It has been shown in [14] that for any $\hat{p} > 0$, the pull-update strategy is guaranteed to satisfy the condition in Theorem 2.

Now, having introduced both the dispatching strategy and the update strategy, we can combine them to obtain different LED policies that are delay optimal in heavy-traffic. For example, we have L-JSQ-Push, L-JSQ-Pull, L-JBA-Push, L-JBA-Pull for heterogeneous servers, as well as L-Pod-Push and L-Pod-Pull for homogeneous servers and for heterogeneous servers satisfy condition Eq. (7).

We end this section by summarizing the contributions of the LED framework. (i) **It covers previous polices.** L-JSQ-Push (with $\hat{p} = 1$) and L-JSQ-Pull are the same as LSQ policies considered in [14], which include the policies developed in both [18] and [19] as special cases. Thus, by Theorems 1 and 2, all these policies are throughput and heavy-traffic delay optimal. (ii) **It allows randomness in dispatching.** The randomness introduced in L-JBA and L-Pod is helpful when dealing with the scenario with an extreme low budget on the message overhead, as discussed next. (iii) **It enables trade-offs between memory and message overhead.** For example, L-Pod-Push and L-Pod-Pull represent good examples that trade memory for low message overhead. That is, if each dispatcher directly uses the traditional Power-of-d without any memory, then at least 4 messages are needed to guarantee delay optimality in heavy-traffic. In contrast, in both L-Pod-Push and L-Pod-Pull, the *worst-case* message overhead is just 1 per arrival. In addition, the message can be further reduced by choosing a smaller value of $\hat{p}$ in the update strategy.

4. Discussion

Before moving to the proofs, we would like to discuss key features and insights about LED, and point out possible refinements on LED.

4.1. Key features of LED

In this section, we highlight the key features of the LED framework, including low message overhead, zero dispatching delay, low computational complexity and appealing performance across various loads.

Low message overhead. It should be noted that the communication overhead occurs only during the update phase in LED policies. For the push-update strategy, the number of messages per arrival is at most $2d$ (d can even be one). For the pull-update strategy, the number of messages per arrival is at most 1. In contrast, JSQ needs $2N$ messages per arrival and Power-of-d needs at least 4 messages per arrival. Although JIQ has a comparative worst-case message overhead as LED policies, it is not stable for heterogeneous servers.

Zero dispatching delay. Another key feature of all LED policies is that there is zero dispatching delay. That is, the dispatcher can immediately route its new arrivals to the chosen server since the decision is made purely based on its local estimations. Moreover, the communication between dispatchers and servers happens only after the decision is made. This is in contrast to typical push-based policies like JSQ and Power-of-d, under which the dispatcher has to wait for the response of sampled servers to make its dispatching decision, resulting in a non-zero dispatching delay.

Low computational complexity. In order to implement LED policies, each dispatcher has to keep an array of size N its local estimations. Such a space requirement is negligible in a modern cluster. Further, the operations required by dispatching strategies of LED policies are very efficient. For example, in order to find the server with the minimal local estimate in L-JSQ, we can keep the array in a min-heap data structure. For L-JBA, we can calculate the average by using an efficient running average algorithm. For the simple L-Pod, it only needs random number generators.

Appealing performance across loads. Although the theoretical delay optimality for the LED framework holds in the heavy-traffic asymptotic regime, the family of LED policies includes efficient policies that significantly outperform alternative low-message overhead policies with the same (or even smaller) amount of communications. For example, if the dispatching strategy adopts L-JSQ in LED, then it reduces to the LSQ policy proposed in [14], which appeals to enjoy good performance over a wide range of traffic loads in different scenarios via extensive simulations.

As mentioned earlier, the class of heavy-traffic delay optimal LED policies is broad and includes flexible choices of different dispatching and update strategies based on different application scenarios. The actual delay performance (except the heavy-load scenario) varies with the particular choice of dispatching strategy or update strategy under different scenarios. Thus, it is not possible to pick one particular LED policy that fits every circumstance, which is also not the focus of this paper. Instead, it would be useful to present some useful insights about the LED framework, as presented in the following. These insights could serve as the guidance on the choice or design of new LED policies.

4.2. Useful insights from LED

The main trait of the LED framework is that only local, possibly delayed and inaccurate information, is used for making the dispatching decision. In the following, we present two useful insights about the use of inaccurate delayed information for load balancing.

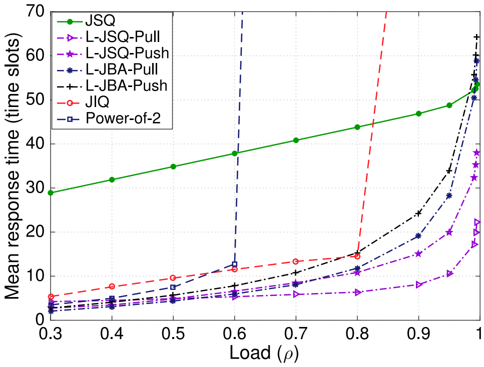

Fig. 1. Inaccurate information could improve performance in multiple-dispatcher case.

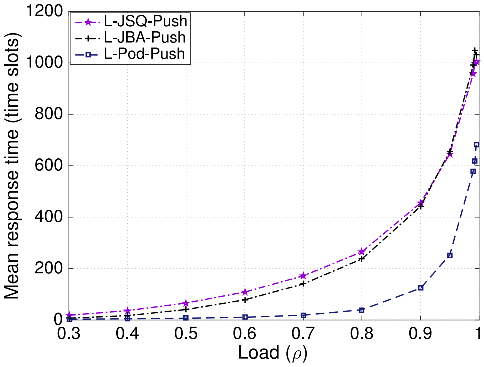

Fig. 2. Randomness is useful for heavily-delayed information.

Inaccurate information can improve performance. A big problem for load balancing with multiple dispatchers is *herd behavior*, which means that arrivals at different dispatchers join the same server. This often leads to a poor delay performance in practice [24]. For example, JSQ used in the case of multiple dispatchers leads to a serious herd behavior since all the dispatchers will route arrivals to the single shortest queue. In contrast, under the LED framework, each dispatcher may believe that a different queue is the shortest according to its own local estimates because these estimates are inaccurate and delayed. Thus, jobs at different dispatchers are sent to different queues that may not have the actual shortest length but still have relatively small queue lengths. This intuition is illustrated by Fig. 1. In particular, we consider a set up with 10 dispatchers and 100 heterogeneous servers. All the LED policies are configured to have the same average message overheads as Power-of-2. It can be seen that the LED policies are not only stable but also achieve a much better performance compared to JSQ, which suffers from the herd behavior in the multiple-dispatcher case.

Randomness is useful for heavily-delayed information. As mentioned earlier, the LED framework provides us with the possibility of exploring load balancing with extremely low message overhead by choosing a small value $\hat{p}$ in the update strategy. As a result, the local information at each dispatcher will only be updated after a long time interval. In this case, if a deterministic dispatching strategy (e.g., L-JSQ) is adopted, it would again incur herd behavior (even for a single dispatcher case) since all the arrivals during the long update interval will join the same queue. This is another motivation for considering L-JBA and L-Pod, which naturally introduce a certain level of randomness and hence help avoid the herd behavior as suggested by [25]. To illustrate this insight, we consider a set up with 10 dispatchers and 100 homogeneous servers. We compare the delay performance of L-JSQ-Push, L-Pod-Push and L-JBA-Push with the update probability set to $\hat{p} = 0.01$ and $d = 2$. As shown in Fig. 2, both L-JBA-Push and L-Pod-Push outperforms L-JSQ-Push, which suffers from herd behavior because of heavily-delayed information.

4.3. Refinements on LED

Our main results suggest that there is a large class of heavy-traffic delay optimal LED policies. On the one hand, it provides us with flexibility to tailor our policy design for different application scenarios with different choices of dispatching and update strategies. On the other hand, it also suggests the need for refinements on LED beyond delay optimality in heavy-traffic. To this end, we introduce two possible directions for refinements.

Degree of queue imbalance. As introduced in [26], *degree of queue imbalance* is a refined metric to further distinguish heavy-traffic delay optimal policies. The idea is that, instead of looking at the average queue length (and hence average delay), the degree of queue imbalance measures the expected difference in queue lengths among the servers. By following the proof of Proposition 5.6 in [26], we can establish that the degree of queue imbalance of all heavy-traffic delay optimal LED policies is $O(\frac{1}{\delta^2 p^4})$. Thus, even though by Theorem 2, any positive δ and p are sufficient for delay optimality in heavy-traffic, a dispatching strategy with smaller δ or an update strategy with a smaller p could affect the performance in practice.

Other asymptotic regimes. In this paper, we focus on the heavy-traffic asymptotic regime where the number of servers is fixed and the load approaches one. As mentioned before, there are also other asymptotic regimes in the analysis of load balancing schemes. One possible direction is to extend the fluid-limit techniques for the large-system regime in [19] to the case of multiple dispatchers and heterogeneous servers. Another alternative regime is the many-server heavy-traffic regime (e.g., Halfin–Whitt regime), which tends to keep a balance between heavy-traffic regime and large-system regime. Studying LED in such a regime is another interesting direction for future work.

5. Proofs

In this paper, we extend the Lyapunov drift-based approach developed in [3] to allow for unbounded supports of arrival and service processes. In particular, we replace the finiteness condition on the drift in [3] by a stochastically dominated condition, as shown in (C2) in Lemma 2. As proved in [27], this weaker condition, combined with a negative drift condition, can still guarantee finite moment bounds. Besides a weaker condition, we also replace the one-step drift with a T-step drift. Formally, we use the following lemma to derive bounded moments in steady state.

Lemma 2. *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*

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

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

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

- *(C2) $|\Delta V(X)| \prec W$ for all t_0 and all $X \in \mathcal{X}$, and $\mathbb{E}\left[e^{\theta W}\right] = D$ is finite for some $\theta > 0$,*

Then $\{V(X(t)), t \geq 0\}$ converges in distribution to a random variable $\overline{V}$ for which there exist a $\theta^* > 0$ and a $C^* < \infty$ such that

$$\mathbb{E}\left[e^{\theta^* \overline{V}}\right] \leq C^*,$$

which directly implies that all the moments of $\overline{V}$ exist and are finite.

5.1. Proof of Theorem 1

To start with, let us first show that the Markov chain $\{Z(t) = (\mathbf{Q}(t), m(t)), t \geq 0\}$ with $m(t) \triangleq (\widetilde{\mathbf{Q}}^1(t), \widetilde{\mathbf{Q}}^2(t), \ldots, \widetilde{\mathbf{Q}}^m(t))$ is irreducible and aperiodic. Let the initial state be $Z(0) = (\mathbf{Q}(0), m(0)) = (0_{1\times N}, 0_{1\times MN})$ and the state space $\mathcal{Z}$ consists of all the states that can be reached from the initial state. Consider any state Z, the queue length vector $\mathbf{Q}$ can reach the initial state with a positive probability since the event that there are no exogenous arrivals and all the offered service is at least one during each time-slot happens with positive probability under our assumptions. Moreover, under the condition for the update strategy given by Eq. (5), the event that $\mathbf{Q}$ remains as the initial state while all $\widetilde{\mathbf{Q}}^m$ reach the initial state happens with a positive probability. Therefore, any state in the state space can reach the initial state, and hence the Markov chain is irreducible. The aperiodicity of the Markov chain comes from the fact that the transition probability from the initial state to itself is positive.

In order to show positive recurrence, we adopt the Foster–Lyapunov theorem. In particular, we consider the following Lyapunov function $W(Z(t)) = \|\mathbf{Q}(t)\|^2 + \sum_{m=1}^{M} \left\|\mathbf{Q}(t) - \widetilde{\mathbf{Q}}^m(t)\right\|_1$, and in the rest of the proof we use $W(t)$ as an abbreviation of $W(Z(t))$ Let $X_n^m(t) \triangleq |Q_n(t) - \widetilde{Q}_n^m(t)|$. The conditional mean drift of $W(t)$ defined as $D(Z(t_0)) \triangleq \mathbb{E}[W(t_0 + T) - W(t_0) \mid Z(t_0)]$ can be decomposed as follows

$$D(Z(t_0)) = D_Q(t_0) + \sum_{m=1}^{M}\sum_{n=1}^{N} D_{X_n^m}(t_0) \tag{8}$$

where

$$D_Q(t_0) \triangleq \mathbb{E}\left[\|\mathbf{Q}(t_0+T)\|^2 - \|\mathbf{Q}(t_0)\|^2 \mid Z(t_0)\right]$$
$$D_{X_n^m}(t_0) \triangleq \mathbb{E}\left[X_n^m(t_0+T) - X_n^m(t_0) \mid Z(t_0)\right]$$

Let us first consider the tern $D_{X_n^m}(t_0)$. Note that for all t_0, m and n

$$\begin{aligned}
&\mathbb{E}\left[X_n^m(t_0+1) \mid Z(t_0)=Z\right]\\
\le&\mathbb{E}\left[(1-\mathcal{I}_n^m(t_0))\left(X_n^m(t_0)+A_n(t_0)+S_n(t_0)\right) \mid Z(t_0)=Z\right]\\
\overset{(a)}{\le}&(1-p)X_n^m(t_0)+\lambda_\Sigma+\mu_{max}
\end{aligned} \tag{9}$$

where (a) follows from the condition in Eq. (5) and $\mu_{max} = \max_n \mu_n$. Then, we have (the time reference t_0 is dropped for simplicity)

$$\begin{aligned}
&D_{X_n^m}(t_0)\\
=&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} X_n^m(t+1) - X_n^m(t) \mid Z(t_0)=Z\right]\\
=&\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\mathbb{E}\left[X_n^m(t+1)-X_n^m(t) \mid Z(t)\right] \mid Z\right]\\
\overset{(a)}{\le}&\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[-pX_n^m(t)+\lambda_\Sigma+\mu_{max} \mid Z\right]\\
\le&-pX_n^m(t_0)+\lambda_\Sigma+\mu_{max},
\end{aligned} \tag{10}$$

where (a) follows from Eq. (9). Let us turn to consider the term $D_Q(t_0)$. By the queue dynamics in Eq. (2),

$$\begin{aligned}
&D_Q(t_0)\\
=&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \|\mathbf{Q}(t+1)\|^2 - \|\mathbf{Q}(t)\|^2 \mid Z(t_0)=Z\right]\\
=&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \|\mathbf{Q}(t)+\mathbf{A}(t)-\mathbf{S}(t)+\mathbf{U}(t)\|^2 - \|\mathbf{Q}(t)\|^2 \mid Z\right]\\
\overset{(a)}{\le}&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \|\mathbf{Q}(t)+\mathbf{A}(t)-\mathbf{S}(t)\|^2 - \|\mathbf{Q}(t)\|^2 \mid Z\right]\\
=&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} 2\langle\mathbf{Q}(t),\mathbf{A}(t)-\mathbf{S}(t)\rangle + \|\mathbf{A}(t)-\mathbf{S}(t)\|^2 \mid Z\right]\\
\overset{(b)}{\le}&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} 2\langle\mathbf{Q}(t),\mathbf{A}(t)-\mathbf{S}(t)\rangle + K \mid Z\right],
\end{aligned} \tag{11}$$

where (a) follows from the facts that $Q_n(t)+A_n(t)-S_n(t)+U_n(t) = \max(Q_n(t)+A_n(t)-S_n(t),0)$ for any $t \ge 0$, and $(\max(a,0))^2 \le a^2$ for any $a \in \mathbb{R}$; (b) holds by our assumption of light-tailed distributions for the total arrival process and each service process in Eq. (3). In particular, we have that the second moments for total arrival process and service process of each server are finite (independent of ϵ), and hence there exists a finite upper bound K which is independent of the load parameter ϵ.

Now, let us continue to work on Eq. (11). In particular, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \langle\mathbf{Q}(t),\mathbf{A}(t)-\mathbf{S}(t)\rangle \mid Z(t_0)=Z\right]\\
=&\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\mathbb{E}\left[\langle\mathbf{Q}(t),\mathbf{A}(t)-\mathbf{S}(t)\rangle \mid Z(t)\right] \mid Z(t_0)=Z\right]
\end{aligned}$$

$$= \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\mathbb{E}\left[\langle \mathbf{Q}(t), \mathbf{A}(t)\rangle \mid Z(t)\right] \mid Z(t_0) = Z\right] \tag{12}$$

$$- \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)\mu_n \mid Z(t_0) = Z\right]. \tag{13}$$

For Eq. (12), we have

$$\begin{aligned}
&\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\mathbb{E}\left[\langle \mathbf{Q}(t), \mathbf{A}(t)\rangle \mid Z(t)\right] \mid Z(t_0) = Z\right]\\
&= \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t) \sum_{m=1}^{M} \mathbb{E}\left[A_n^m(t) \mid Z(t)\right] \mid Z\right]\\
&= \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t) \sum_{m=1}^{M} P_n^m(t)\lambda_m \mid Z(t_0) = Z\right]\\
&\overset{(a)}{=} \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t) \sum_{m=1}^{M} \left(\beta_n^m(t) + \frac{\mu_n}{\mu_\Sigma}\right)\lambda_m \mid Z(t_0) = Z\right],
\end{aligned}$$

where (a) follows from the definition of $\beta_n^m(t)$. Then, it can be further simplified as follows.

$$\begin{aligned}
&\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\mathbb{E}\left[\langle \mathbf{Q}(t), \mathbf{A}(t)\rangle \mid Z(t)\right] \mid Z(t_0) = Z\right]\\
&= \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t) \sum_{m=1}^{M} \beta_n^m(t)\lambda_m \mid Z\right]\\
&\quad + \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)\mu_n \mid Z\right] - \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)\frac{\epsilon\mu_n}{\mu_\Sigma} \mid Z\right].
\end{aligned} \tag{14}$$

Combining Eqs. (12), (13) and (14), yields

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \langle \mathbf{Q}(t), \mathbf{A}(t) - \mathbf{S}(t)\rangle \mid Z(t_0) = Z\right]\\
&= \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left(Q_n(t) - \widetilde{Q}_n^m(t) + \widetilde{Q}_n^m(t)\right)\beta_n^m(t)\lambda_m \mid Z\right]\\
&\quad - \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)\frac{\epsilon\mu_n}{\mu_\Sigma} \mid Z\right].
\end{aligned}$$

The RHS of the above equation can be further written as

$$\begin{aligned}
\text{RHS} = &\underbrace{\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left(Q_n(t) - \widetilde{Q}_n^m(t)\right)\beta_n^m(t)\lambda_m \mid Z\right]}_{\mathcal{T}_1}\\
&+ \underbrace{\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \widetilde{Q}_n^m(t)\beta_n^m(t)\lambda_m \mid Z\right]}_{\mathcal{T}_2} - \underbrace{\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_n(t)\frac{\epsilon\mu_n}{\mu_\Sigma} \mid Z\right]}_{\mathcal{T}_3}.
\end{aligned}$$

We are going to handle each term one by one. To upper bound $\mathcal{T}_1$, we use the following result on $X_n^m(t) = |Q_n(t) - \widetilde{Q}_n^m(t)|$.

Lemma 3. *Under the condition given by Eq. (5), for any t_0 and $Z(t_0)$, there exist a finite T_1 independent of ϵ and a finite constant L that is only a function of p and μ_Σ, such that for all $T \geq T_1$*

$$\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} X_m^n(t) \mid Z(t_0) = Z\right] \leq LT$$

holds for all m and n.

Proof. See Appendix A. □

By using Lemma 3 with $T \geq T_1$, we have

$$\mathcal{T}_1 \leq \lambda_\Sigma \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left|Q_n(t) - \widetilde{Q}_n^m(t)\right| \mid Z\right] \leq \lambda_\Sigma MNLT. \tag{15}$$

For $\mathcal{T}_2$, we have

$$\mathcal{T}_2 \overset{(a)}{=} \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \widetilde{Q}_{\sigma_t(n)}^m(t)\Delta_n^m(t)\lambda_m \mid Z\right] \overset{(b)}{\leq} 0, \tag{16}$$

where (a) comes from the definition of dispatching preference vector $\Delta^m(t)$; (b) holds due to the δ-tilted sum condition. In particular, we have the following decomposition of the term in the expectation.

$$\begin{aligned}
&\mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \widetilde{Q}_{\sigma_t(n)}^m(t)\Delta_n^m(t)\lambda_m \mid Z\right]\\
=&\mathbb{E}\left[\sum_{m=1}^{M}\left(\widetilde{Q}_{\sigma_t(1)}^m(t)\sum_{n=1}^{N}\Delta_n^m(t)\right) \mid Z\right] \qquad (17)\\
&+\mathbb{E}\left[\sum_{m=1}^{M}\left(\sum_{k=2}^{N}\left(\sum_{n=k}^{N}\Delta_n^m(t)\right)(\widetilde{Q}_{\sigma_t(k)}^m(t) - \widetilde{Q}_{\sigma_t(k-1)}^m(t))\right) \mid Z\right]. \qquad (18)
\end{aligned}$$

Note that Eq. (17) is zero since $\sum_{n=1}^N \Delta_n^m(t)$ by the definition of dispatcher preference $\Delta^m(t)$. Moreover, by the δ-tilted sum condition, we have $\sum_{n=k}^N \Delta_n^m(t) = 0 - \sum_{n=1}^{k-1}\Delta_n^m(t) \leq -\delta \leq 0$ for all $k \geq 2$. Hence, Eq. (18) is less than or equal to zero since $\widetilde{Q}_{\sigma_t(k)}^m(t) - \widetilde{Q}_{\sigma_t(k-1)}^m(t) \geq 0$ by the definition of the permutation $\sigma_t(\cdot)$.

For $\mathcal{T}_3$, we have

$$\mathcal{T}_3 \geq \frac{\epsilon\mu_{min}}{\mu_\Sigma}\left\|\mathbf{Q}(t_0)\right\|_1, \tag{19}$$

where $\mu_{min} = \min_n \mu_n$.

Now, combining Eqs. (15), (16) and (19), yields

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\langle\mathbf{Q}(t), \mathbf{A}(t) - \mathbf{S}(t)\rangle \mid Z(t_0) = Z\right]\\
&\leq -\frac{\epsilon\mu_{min}}{\mu_\Sigma}\left\|\mathbf{Q}(t_0)\right\|_1 + \lambda_\Sigma MNLT.
\end{aligned}$$

Substituting the result above back into Eq. (11), yields

$$D_Q(t_0) \leq -2\frac{\epsilon\mu_{min}}{\mu_\Sigma}\left\|\mathbf{Q}(t_0)\right\|_1 + 2\lambda_\Sigma MNLT + KT. \tag{20}$$

Now, we are ready to substitute Eqs. (10) and (20) back into Eq. (8). As a result, we have

$$\begin{aligned}
D(Z(t_0)) \leq& -2\frac{\epsilon\mu_{min}}{\mu_\Sigma}\left\|\mathbf{Q}(t_0)\right\|_1 - p\sum_{m=1}^{M}\sum_{n=1}^{N}X_n^m(t_0)\\
&+ 2\lambda_\Sigma MNLT + KT + \lambda_\Sigma + \mu_{max}\\
\overset{(a)}{\leq}& -\xi\left(\left\|\mathbf{Q}(t_0)\right\|_1 + \sum_{m=1}^{M}\sum_{n=1}^{N}\left|Q_n(t_0) - \widetilde{Q}_n^m(t_0)\right|\right) + K_1,
\end{aligned}$$

where in (a) $\xi = \min(2\frac{\epsilon\mu_{min}}{\mu_\Sigma}, p)$ and $K_1 \triangleq 2\lambda_\Sigma MNLT + KT + \lambda_\Sigma + \mu_{max}$. Pick any $\alpha > 0$ and let

$$\mathcal{B} \triangleq \{Z \in \mathcal{Z} : \|\mathbf{Q}(t_0)\|_1 + \sum_{m=1}^{M}\sum_{n=1}^{N} |Q_n(t_0) - \widetilde{Q}_n^m(t_0)| \leq \frac{K_1+\alpha}{\xi}\}.$$

Then, $\mathcal{B}$ is a finite subset. For any $Z \in \mathcal{B}^c$, $D(Z) \leq -\alpha$, and for any $Z \in \mathcal{B}$, $D(Z) \leq K_1$. By Foster–Lyapunov theorem, we have established positive recurrence.

Having shown that the Markov chain $\{Z(t), t \geq 0\}$ is ergodic, we are left with the task of showing that all the moments are finite in steady-state. In order to do so, we use Lemma 2. In particular, we choose the Lyapunov function as $V(Z^{(\epsilon)}) = \left\|\mathbf{Q}^{(\epsilon)}\right\|$ and then verify the two conditions. In the following, the superscript $^{(\epsilon)}$ will be omitted for ease of notations. To verify condition (C2), we have

$$\begin{aligned}
|\Delta V(Z)| &= |\|\mathbf{Q}(t_0+T)\| - \|\mathbf{Q}(t_0)\|\,| \mathcal{I}(Z(t_0)=Z) \\
&\overset{(a)}{\leq} \|\mathbf{Q}(t_0+T) - \mathbf{Q}(t_0)\|\, \mathcal{I}(Z(t_0)=Z) \\
&\leq \sum_{t=t_0}^{t_0+T-1} \|\mathbf{Q}(t+1) - \mathbf{Q}(t)\|\, \mathcal{I}(Z(t_0)=Z) \\
&\leq \sum_{t=t_0}^{t_0+T-1} \|\mathbf{A}(t) - \mathbf{S}(t) + \mathbf{U}(t)\|\, \mathcal{I}(Z(t_0)=Z) \\
&\overset{(b)}{\leq} \sum_{t=t_0}^{t_0+T-1} (\|\mathbf{A}(t)\| + 2\,\|\mathbf{S}(t)\|)\, \mathcal{I}(Z(t_0)=Z),
\end{aligned} \tag{21}$$

where (a) holds since $|\|\mathbf{x}\| - \|\mathbf{y}\|\,| \leq \|\mathbf{x}-\mathbf{y}\|$ for each $\mathbf{x}, \mathbf{y}$ in $\mathbb{R}^N$. (b) follows from triangle inequality and the fact that $U_n(t) \leq S_n(t)$ for all t and t. Then, by our assumptions of light-tailed distributions for both total arrival and service processes, there exists a random variable W such that $|\Delta V(X)| \prec W$ for all t_0 and all $X \in \mathcal{X}$, and $\mathbb{E}\left[e^{\theta W}\right] = D$ is finite for some $\theta > 0$, which verifies (C2).

For (C1), we have

$$\begin{aligned}
&\mathbb{E}\left[\Delta V(Z) \mid Z(t_0)=Z\right] \\
=&\mathbb{E}\left[\|\mathbf{Q}(t_0+T)\| - \|\mathbf{Q}(t_0)\| \mid Z(t_0)=Z\right] \\
=&\mathbb{E}\left[\sqrt{\|\mathbf{Q}(t_0+T)\|^2} - \sqrt{\|\mathbf{Q}(t_0)\|^2} \mid Z(t_0)=Z\right] \\
\overset{(a)}{\leq}&\frac{1}{2\,\|\mathbf{Q}(t_0)\|}\mathbb{E}\left[\|\mathbf{Q}(t_0+T)\|^2 - \|\mathbf{Q}(t_0)\|^2 \mid Z(t_0)=Z\right] \\
\overset{(b)}{\leq}& -\epsilon\frac{\mu_{min}}{\mu_\Sigma} + \frac{2\lambda_\Sigma MNLT + KT}{2\,\|\mathbf{Q}(t_0)\|},
\end{aligned}$$

where (a) follows from the fact that $f(x) = \sqrt{x}$ is concave; (b) comes from Eq. (20). Thus, condition (C1) is valid and hence the proof of Theorem 1 is complete.

5.2. Proof of Theorem 2

In order to prove the result, we need two intermediate results. One is called *state-space collapse* as stated in Proposition 2, which is the key ingredient for establishing heavy traffic delay optimality. Roughly speaking, it means that the multi-dimensional space for the queue length vector reduces to one dimension in the sense that the deviation from the line (on which all the queue lengths are equal) is bounded by a constant, independent of ϵ. Another intermediate result is concerned with unused service. Based on these two intermediate results, we can prove heavy-traffic delay optimality. We omit the time reference t_0 for simplicity when necessary.

Proposition 2. *Under the conditions in Theorem 2, then we have that $\mathbf{Q}_\perp$ is bounded in the sense that in steady state there exist finite constants $\{L_r, r \in \mathbb{N}\}$ independent of ϵ such that*

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

for all $\epsilon \in (0, \epsilon_0)$ and $r \in \mathbb{N}$, where $\mathbf{Q}_\perp = \mathbf{Q} - \langle \mathbf{Q}, \mathbf{c}\rangle \mathbf{c}$ is the perpendicular component of $\mathbf{Q}$ with respect to the line $\mathbf{c} = \frac{1}{\sqrt{N}}(1, 1, \dots, 1)$.

Proof. It suffices to show that $V_\perp(Z^{(\epsilon)}) \triangleq \left\| \mathbf{Q}_\perp^{(\epsilon)} \right\|$ satisfies the conditions (C1) and (C2) in Lemma 2. Let us first consider conditions (C2). In particular, we have

$$\begin{aligned}
&|\Delta V_\perp(Z)| \\
=& | \|\mathbf{Q}_\perp(t_0+T)\| - \|\mathbf{Q}_\perp(t_0)\| \, | \mathcal{I}(Z(t_0)=Z) \\
\overset{(a)}{\le}& \|\mathbf{Q}_\perp(t_0+T) - \mathbf{Q}_\perp(t_0)\| \, \mathcal{I}(Z(t_0)=Z) \\
=& \left\| \mathbf{Q}(t_0+T) - \mathbf{Q}_\parallel(t_0+T) - \mathbf{Q}(t_0) + \mathbf{Q}_\parallel(t_0) \right\| \mathcal{I}(Z(t_0)=Z) \\
\overset{(b)}{\le}& \|\mathbf{Q}(t_0+T) - \mathbf{Q}(t_0)\| + \left\| \mathbf{Q}_\parallel(t_0+T) - \mathbf{Q}_\parallel(t_0) \right\| \mathcal{I}(Z(t_0)=Z) \\
\overset{(c)}{\le}& 2 \, \|\mathbf{Q}(t_0+T) - \mathbf{Q}(t_0)\| \, \mathcal{I}(Z(t_0)=Z) \\
\overset{(d)}{\le}& 2 \sum_{t=t_0}^{t_0+T-1} \left(\|\mathbf{A}(t)\| + 2 \, \|\mathbf{S}(t)\| \right) \mathcal{I}(Z(t_0)=Z)
\end{aligned} \tag{22}$$

where the inequality (a) follows from the fact that $| \|\mathbf{x}\| - \|\mathbf{y}\| \, | \le \|\mathbf{x}-\mathbf{y}\|$ holds for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$; inequality (b) follows from triangle inequality; (c) holds due to the non-expansive property of projection to a convex set; (d) follows from Eq. (21). Then by our assumptions of light-tailed distributions for both total arrival and service processes, there exists a random variable W such that $|\Delta V_\perp(X)| \prec W$ for all t_0 and all $X \in \mathcal{X}$, and $\mathbb{E}\left[e^{\theta W}\right] = D$ is finite for some $\theta > 0$, which verifies (C2).

Let us turn to condition (C1). By the proof of Lemma 3.6 in [8], it suffices to establish the following result in order to verify (C1). That is, there exist $T > 0$, $K_2 \ge 0$ and $\eta > 0$ that are all independent of ϵ, such that for all t_0 and $Z \in \mathcal{Z}$

$$\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \langle \mathbf{Q}_\perp(t), \mathbf{A}(t) - \mathbf{S}(t) \rangle \mid Z(t_0) = Z \right] \le -\eta \, \|\mathbf{Q}_\perp\| + K_2 \tag{23}$$

holds for all $\epsilon \in (0, \epsilon_0)$. Note that

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \langle \mathbf{Q}_\perp(t), \mathbf{A}(t) - \mathbf{S}(t) \rangle \mid Z(t_0) = Z \right] \\
\overset{(a)}{=}& \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\mathbb{E}\left[\langle \mathbf{Q}_\perp(t), \mathbf{A}(t) \rangle \mid Z(t) \right] \mid Z(t_0) = Z \right] \qquad (24) \\
&- \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_n \mu_n Q_{\perp,n}(t) \mid Z(t_0) = Z \right], \qquad (25)
\end{aligned}$$

where (a) follows from the tower property of conditional expectation and the fact that $A(t)$ is independent of $Z(t_0)$ given $Z(t)$. Moreover, $Q_{\perp,n}(t)$ denotes the nth component of the vector $\mathbf{Q}_\perp(t)$. Now let us first focus on Eq. (24).

$$\begin{aligned}
&\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\mathbb{E}\left[\langle \mathbf{Q}_\perp(t), \mathbf{A}(t) \rangle \mid Z(t) \right] \mid Z(t_0) = Z \right] \\
=& \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_{\perp,n}(t) \sum_{m=1}^{M} \beta_n^m(t) \lambda_m \mid Z \right] \\
&+ \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_{\perp,n}(t) \sum_{m=1}^{M} \frac{\mu_n}{\mu_\Sigma} \left(\mu_\Sigma - \epsilon \right) p_m \mid Z \right].
\end{aligned}$$

Combining the result above with Eq. (25), yields

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \langle \mathbf{Q}_\perp(t), \mathbf{A}(t) - \mathbf{S}(t) \rangle \mid Z(t_0) = Z \right] \\
=& \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_{\perp,n}(t) \sum_{m=1}^{M} \beta_n^m(t) \lambda_m \mid Z \right] \qquad (26) \\
&+ \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_{\perp,n}(t) \frac{-\epsilon \mu_n}{\mu_\Sigma} \mid Z \right]. \qquad (27)
\end{aligned}$$

Note that by definition $Q_{\perp,n}(t) = Q_n(t) - Q_{\text{avg}}(t)$, in which $Q_{\text{avg}}(t)$ is the average queue length among N queues at the beginning of time-slot t. Moreover, $Q_{\perp,n}(t)$ can be written as

$$Q_{\perp,n}(t) = Q_n(t) - \widetilde{Q}_n^m(t) + \widetilde{Q}_n^m(t) - \bar{Q}^m(t) + \bar{Q}^m(t) - Q_{\text{avg}}(t) \tag{28}$$

for all m and t, in which $\bar{Q}^m(t) \triangleq \frac{1}{N}\sum_{n=1}^{N} \widetilde{Q}_n^m(t)$, i.e., the average queue length estimated by dispatcher m at the beginning of time-slot t. By utilizing Eq. (28), Eq. (26) can be written as

$$\begin{aligned}
&\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_{\perp,n}(t) \sum_{m=1}^{M} \beta_n^m(t)\lambda_m \mid Z\right] \\
&= \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left(\widetilde{Q}_n^m(t) - \bar{Q}^m(t)\right) \beta_n^m(t)\lambda_m \mid Z\right] && (29)\\
&\quad + \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left(Q_n(t) - \widetilde{Q}_n^m(t)\right) \beta_n^m(t)\lambda_m \mid Z\right] && (30)\\
&\quad + \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left(\bar{Q}^m(t) - Q_{\text{avg}}(t)\right) \beta_n^m(t)\lambda_m \mid Z\right]. && (31)
\end{aligned}$$

Our main task now is to upper bound each term above. Let us start with Eq. (29). In particular, we can bound it by using the following result.

Lemma 4. *There exist finite positive constants η and C such that*

$$\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left(\widetilde{Q}_n^m(t) - \bar{Q}^m(t)\right) \beta_n^m(t)\lambda_m \mid Z\right] \le -\eta \left\|\mathbf{Q}_{\perp}(t_0)\right\| + C$$

holds for all $T \ge 3$, in which $\eta = \frac{\lambda_\Sigma \delta p^2}{\sqrt{N}}$ and $C = 3(\mu_\Sigma)^2 p^2$.

Proof. See Appendix B □

For Eqs. (30) and (31), we can bound both of them by using the result in Lemma 3, respectively. In particular, for Eq. (30), we have

$$\begin{aligned}
&\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left(Q_n(t) - \widetilde{Q}_n^m(t)\right) \beta_n^m(t)\lambda_m \mid Z\right] \\
&\le \lambda_\Sigma \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left|Q_n(t) - \widetilde{Q}_n^m(t)\right| \mid Z\right] \\
&\le \mu_\Sigma MNLT. && (32)
\end{aligned}$$

For Eq. (31), we have

$$\begin{aligned}
&\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left(\bar{Q}^m(t) - Q_{\text{avg}}(t)\right) \beta_n^m(t)\lambda_m \mid Z\right] \\
&= \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \left(\frac{1}{N}\sum_{n=1}^{N} \left(\widetilde{Q}_n^m(t) - Q_n(t)\right)\right) \beta_n^m(t)\lambda_m \mid Z\right] \\
&\le \lambda_\Sigma \sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{m=1}^{M} \frac{1}{N}\sum_{n=1}^{N} \left|\widetilde{Q}_n^m(t) - Q_n(t)\right| \mid Z\right] \\
&\le \mu_\Sigma MNLT. && (33)
\end{aligned}$$

We have obtained bounds for Eqs. (29)–(31). Let us turn to focus on Eq. (27), which can be upper bounded by the following result.

Lemma 5. *For any t_0 and Z,*

$$\sum_{t=t_0}^{t_0+T-1} \mathbb{E}\left[\sum_{n=1}^{N} Q_{\perp,n}(t)\frac{-\epsilon\mu_n}{\mu_\Sigma} \mid Z(t_0)=Z\right] \le \epsilon\sqrt{N}T\left\|\mathbf{Q}(t_0)\right\| + K_3,$$

where k_3 is a finite constant independent of ϵ.

Proof. See Appendix C □

Now, we are ready to bound the left-hand-side of Eq. (23) by using the bounds for both Eqs. (26) and (27). In particular, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\langle \mathbf{Q}_\perp(t),\mathbf{A}(t)-\mathbf{S}(t)\rangle \mid Z(t_0)=Z\right]\\
&\le -\frac{\lambda_\Sigma\delta p^2}{\sqrt{N}}\left\|\mathbf{Q}_\perp(t_0)\right\| + C + 2\mu_\Sigma MNLT + \epsilon\sqrt{N}T\left\|\mathbf{Q}(t_0)\right\| + K_3\\
&\overset{(a)}{=}\left(T\epsilon - \frac{\lambda_\Sigma\delta p^2}{N}\right)\sqrt{N}\left\|\mathbf{Q}_\perp(t_0)\right\| + K_2\\
&\le -\frac{\mu_\Sigma\delta p^2}{2\sqrt{N}}\left\|\mathbf{Q}_\perp(t_0)\right\| + K_2, \quad \forall\epsilon < \frac{\mu_\Sigma\delta p^2}{2NT+2\delta p^2}
\end{aligned} \tag{34}$$

where (a) follows from $K_2 = C + 2\mu_\Sigma MNLT + K_3$, which is independent of ϵ. Hence, this verifies condition (C1) with $\eta = \frac{\mu_\Sigma\delta p^2}{2\sqrt{N}}$, which is also independent of ϵ. Combined with condition (C2), we have finished the proof of Proposition 2. □

Having proved the state-space collapse result, we turn to prove another intermediate result regarding unused service, as stated in the following lemma. In words, this lemma says that in heavy traffic unused service tends to be zero.

Lemma 6. *Under any LED policy, we have*

$$\lim_{\epsilon\downarrow 0}\mathbb{E}\left[\left\|\overline{\mathbf{U}}^{(\epsilon)}\right\|^2\right] = 0.$$

Proof. First, we would like to show that under any LED policy,

$$\mathbb{E}\left[\left\|\overline{\mathbf{U}}^{(\epsilon)}\right\|_1\right] = \epsilon. \tag{35}$$

To see this, we consider the Lyapunov function $W_1(Z(t)) = \|\mathbf{Q}(t)\|_1$. Since LED is throughput optimal with all the moments being finite, we have that the mean drift of $W_1(Z(t))$ in steady-state is zero. Then, we have

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

which directly implies the result in Eq. (35).

Now let us fix $n \in \mathcal{N}$, we have for any $t \ge 0$ and constant S'

$$\begin{aligned}
U_n^2(t) &\le U_n(t)S_n(t)\\
&= U_n(t)S_n(t)\mathcal{I}\left(S_n(t)\le S'\right) + U_n(t)S_n(t)\mathcal{I}\left(S_n(t) > S'\right)\\
&\le U_n(t)S' + S_n^2(t)\mathcal{I}\left(S_n(t) > S'\right).
\end{aligned}$$

In steady state, we have

$$\begin{aligned}
\mathbb{E}\left[\overline{U}_n^2\right] &\le \mathbb{E}\left[\overline{U}_n\right]S' + \mathbb{E}\left[S_n^2(\infty)\mathcal{I}\left(S_n(\infty) > S'\right)\right]\\
&\overset{(a)}{\le} \epsilon S' + \mathbb{E}\left[S_n^2(0)\mathcal{I}\left(S_n(0) > S'\right)\right]\\
&\overset{(b)}{\le} \epsilon S' + \beta,
\end{aligned}$$

where (a) follows from the fact that $\mathbb{E}\left[\|\overline{\mathbf{U}}^{(\epsilon)}\|_1\right] = \epsilon$ and service process is *i.i.d.*; in (b), we choose S' such that $\mathbb{E}\left[S_n^2(0)\mathcal{I}\left(S_n(0) > S'\right)\right] \le \beta$, which is possible by the exponential decay rate of $S_n(0)$ under the light-tailed assumption.

Thus, we have

$$\lim_{\epsilon\downarrow 0}\mathbb{E}\left[\overline{U}_n^2\right]\leq\beta,$$

for any $\beta>0$. Hence, we have $\lim_{\epsilon\downarrow 0}\mathbb{E}\left[\overline{U}_n^2\right]=0$ for each n, which directly implies our result. □

Now, we are prepared to show that under the conditions in Theorem 2, the system achieves optimal delay in heavy traffic. More specifically, by Lemma 3 in [28], we need only to verify the following condition.

$$\lim_{\epsilon\downarrow 0}\mathbb{E}\left[\left\|\overline{\mathbf{Q}}^{(\epsilon)}(t+1)\right\|_1\left\|\overline{\mathbf{U}}^{(\epsilon)}(t)\right\|_1\right]=0. \tag{36}$$

Let us define $\overline{B}^{(\epsilon)}\triangleq\mathbb{E}\left[\left\|\overline{\mathbf{Q}}^{(\epsilon)}(t+1)\right\|_1\left\|\overline{\mathbf{U}}^{(\epsilon)}(t)\right\|_1\right]$. We can bound it as follows:

$$\begin{aligned}\overline{B}^{(\epsilon)}&\overset{(a)}{=}N\mathbb{E}\left[\langle\overline{\mathbf{U}}^{(\epsilon)}(t),-\overline{\mathbf{Q}}_{\perp}^{(\epsilon)}(t+1)\rangle\right]\\&\overset{(b)}{\leq}N\sqrt{\mathbb{E}\left[\left\|\overline{\mathbf{U}}^{(\epsilon)}\right\|^2\right]\mathbb{E}\left[\left\|\overline{\mathbf{Q}}_{\perp}^{(\epsilon)}(t+1)\right\|^2\right]}\\&\overset{(c)}{=}N\sqrt{\mathbb{E}\left[\left\|\overline{\mathbf{U}}^{(\epsilon)}\right\|^2\right]\mathbb{E}\left[\left\|\overline{\mathbf{Q}}_{\perp}^{(\epsilon)}(t)\right\|^2\right]}\\&\overset{(d)}{\leq}N\sqrt{\mathbb{E}\left[\left\|\overline{\mathbf{U}}^{(\epsilon)}\right\|^2\right]L_2},\end{aligned}$$

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 $t\geq 0$ and the definition of $\mathbf{Q}_{\perp}$; the inequality (b) holds due to Cauchy–Schwartz inequality; the equality (c) is true since the distributions of $\overline{\mathbf{Q}}_{\perp}^{(\epsilon)}(t+1)$ and $\overline{\mathbf{Q}}_{\perp}^{(\epsilon)}(t)$ are the same in steady state; (d) follows from the state-space collapse result in Proposition 2. Finally, by Lemma 6 and the fact that L_2 is independent of ϵ, we have $\lim_{\epsilon\to 0}\overline{B}^{(\epsilon)}=0$, which finishes our proof.

6. Conclusion

We have introduced the Local-Estimation-Driven (LED) framework for load balancing policies in possibly heterogeneous systems with multiple dispatchers. Under this framework, each dispatcher keeps local and possibly outdated estimates of the queue lengths for all the servers, and makes its dispatching decision only based on these local estimates. Communication between dispatchers and servers is only used to update the local estimates. We have established sufficient conditions for LED policies to achieve both throughput optimality and delay optimality in heavy traffic. These sufficient conditions not only establish delay optimality for many previous local-memory based policies, but also enable us to tailor the design of new delay optimal policies based on different application requirements. The heavy-traffic delay optimality of LED policies also resolves a recent open problem on the development of load balancing schemes that have only access to delayed information.

In future work, it will be interesting to investigate LED framework in other asymptotic regimes, e.g., the large-system regime and the many-server heavy-traffic regime.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: S. Theja Maguluri, Georgia Institute of Technology, Atlanta, Georgia, United States C.H. Xia, OHIO STATE UNIVERSITY, Columbus, Ohio, United States.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.peva.2020.102146.

References

[1] R.R. Weber, On the optimal assignment of customers to parallel servers, J. Appl. Probab. (1978) 406–413.

[2] G.J. Foschini, J. Salz, A basic dynamic routing problem and diffusion, IEEE Trans. Commun. 26 (3) (1978) 320–327.

[3] A. Eryilmaz, R. Srikant, Asymptotically tight steady-state queue length bounds implied by drift conditions, Queueing Syst. 72 (3–4) (2012) 311–359.

[4] M. Mitzenmacher, The power of two choices in randomized load balancing, IEEE Trans. Parallel Distrib. Syst. 12 (10) (2001) 1094–1104.

[5] S.T. Maguluri, R. Srikant, L. Ying, Heavy traffic optimal resource allocation algorithms for cloud computing clusters, Perform. Eval. 81 (2014) 20–39.

[6] Y. Lu, Q. Xie, G. Kliot, A. Geller, J.R. Larus, A. Greenberg, Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services, Perform. Eval. 68 (11) (2011) 1056–1071.

[7] A.L. Stolyar, Pull-based load distribution in large-scale heterogeneous service systems, Queueing Syst. 80 (4) (2015) 341–361.

[8] X. Zhou, F. Wu, J. Tan, Y. Sun, N. Shroff, Designing low-complexity heavy-traffic delay-optimal load balancing schemes: Theory to algorithms, Proc. ACM Meas. Anal. Comput. Syst. 1 (2) (2017) 39.

[9] X. Zhou, J. Tan, N. Shroff, Heavy-traffic delay optimality in pull-based load balancing systems: necessary and sufficient conditions, Proc. ACM Meas. Anal. Comput. Syst. 2 (3) (2018) 1–33.

[10] M. Mitzenmacher, Analyzing distributed join-idle-queue: A fluid limit approach, in: Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on, IEEE, 2016, pp. 312–318.

[11] A.L. Stolyar, Pull-based load distribution among heterogeneous parallel servers: the case of multiple routers, Queueing Syst. 85 (1–2) (2017) 31–65.

[12] R. Govindan, I. Minei, M. Kallahalla, B. Koley, A. Vahdat, Evolve or die: High-availability design principles drawn from googles network infrastructure, in: Proceedings of the 2016 ACM SIGCOMM Conference, 2016, pp. 58–72.

[13] P. Shuff, Building a billion user load balancer, 2016.

[14] S. Vargaftik, I. Keslassy, A. Orda, Lsq: Load balancing in large-scale heterogeneous systems with multiple dispatchers, IEEE/ACM Trans. Netw. (2020).

[15] D. Lipshutz, Open problem—load balancing using delayed information, Stoch. Syst. 9 (3) (2019) 305–306.

[16] W. Winston, Optimality of the shortest line discipline, J. Appl. Probab. 14 (1) (1977) 181–189.

[17] M. van der Boor, S. Borst, J. van Leeuwaarden, Load balancing in large-scale systems with multiple dispatchers, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017, pp. 1–9.

[18] J. Anselmi, F. Dufour, Power-of-*d*-choices with memory: Fluid limit and optimality, 2018, arXiv preprint arXiv:1802.06566.

[19] M. van der Boor, S. Borst, J. van Leeuwaarden, Hyper-scalable JSQ with sparse feedback, Proc. ACM Meas. Anal. Comput. Syst. 3 (1) (2019) 1–37.

[20] W. Wang, K. Zhu, L. Ying, J. Tan, L. Zhang, Maptask scheduling in mapreduce with data locality: Throughput and heavy-traffic optimality, IEEE/ACM Trans. Netw. 24 (1) (2016) 190–203.

[21] Q. Xie, A. Yekkehkhany, Y. Lu, Scheduling with multi-level data locality: Throughput and heavy-traffic optimality, in: Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2016, pp. 1–9.

[22] Q. Xie, Y. Lu, Priority algorithm for near-data scheduling: Throughput and heavy-traffic optimality, in: Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2015, pp. 963–972.

[23] D. Hurtado-Lange, S.T. Maguluri, Throughput and delay optimality of power-of-d choices in inhomogeneous load balancing systems, 2020, arXiv preprint arXiv:2004.00538.

[24] L. Suresh, M. Canini, S. Schmid, A. Feldmann, C3: Cutting tail latency in cloud data stores via adaptive replica selection, in: Proceedings of the 2015 USENIX NSDI Conference, 2015, pp. 513–527.

[25] M. Mitzenmacher, How useful is old information?, IEEE Trans. Parallel Distrib. Syst. 11 (1) (2000) 6–20.

[26] X. Zhou, F. Wu, J. Tan, K. Srinivasan, N. Shroff, Degree of queue imbalance: Overcoming the limitation of heavy-traffic delay optimality in load balancing systems, Proc. ACM Meas. Anal. Comput. Syst. 2 (1) (2018) 1–41.

[27] B. Hajek, Hitting-time and occupation-time bounds implied by drift analysis with applications, Adv. Appl. Probab. (1982) 502–525.

[28] X. Zhou, J. Tan, N. Shroff, Flexible load balancing with multi-dimensional state-space collapse: throughput and heavy-traffic delay optimality, Perform. Eval. 127 (2018) 176–193.