

Optimal Load Balancing with Locality Constraints

WENTAO WENG, Institute for Interdisciplinary Information Sciences, Tsinghua University, China

XINGYU ZHOU, ECE, The Ohio State University, USA

R. SRIKANT, CSL and ECE, University of Illinois at Urbana-Champaign, USA

Applications in cloud platforms motivate the study of efficient load balancing under job-server constraints and server heterogeneity. In this paper, we study load balancing on a bipartite graph where left nodes correspond to job types and right nodes correspond to servers, with each edge indicating that a job type can be served by a server. Thus edges represent locality constraints, i.e., an arbitrary job can only be served at servers which contain certain data and/or machine learning (ML) models. Servers in this system can have heterogeneous service rates. In this setting, we investigate the performance of two policies named Join-the-Fastest-of-the-Shortest-Queue (JFSQ) and Join-the-Fastest-of-the-Idle-Queue (JFIQ), which are simple variants of Join-the-Shortest-Queue and Join-the-Idle-Queue, where ties are broken in favor of the fastest servers. Under a “well-connected” graph condition, we show that JFSQ and JFIQ are asymptotically optimal in the mean response time when the number of servers goes to infinity. In addition to asymptotic optimality, we also obtain upper bounds on the mean response time for finite-size systems. We further show that the well-connectedness condition can be satisfied by a random bipartite graph construction with relatively sparse connectivity.

CCS Concepts: • **Mathematics of computing** → **Queueing theory**; **Markov processes**; • **Networks** → **Network performance analysis**; **Cloud computing**.

Additional Key Words and Phrases: cloud computing; load balancing; asymptotic optimality; delay performance

ACM Reference Format:

Wentao Weng, Xingyu Zhou, and R. Srikant. 2020. Optimal Load Balancing with Locality Constraints. *Proc. ACM Meas. Anal. Comput. Syst.* 4, 3, Article 45 (December 2020), 37 pages. <https://doi.org/10.1145/3428330>

1 INTRODUCTION

Many applications that use data centers, cloud computing systems and other data analytic platforms, including Web search engines [23], cloud computing service [1], large-scale data processing [13], and cloud storage have extremely stringent latency requirements. Ultra low latency guarantees in these applications not only provide smooth user experience, but help improve company profits [12].

A key component for achieving a fast response in the aforementioned systems are load balancing algorithms, which are responsible for dispatching jobs to parallel servers. Motivated by the demanding requirement of a low latency, there has been a line of recent research that aims to design smart load balancing algorithms with delay performance guarantees. They often focus on the classical load balancing model, where there are N identical servers with exponential service times and a

Authors' addresses: Wentao Weng, wwt17@mails.tsinghua.edu.cn, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, 100084, China; Xingyu Zhou, zhou.2055@osu.edu, ECE, The Ohio State University, OH, 43210, USA; R. Srikant, rsrikant@illinois.edu, CSL and ECE, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

2476-1249/2020/12-ART45 \$15.00

<https://doi.org/10.1145/3428330>

dispatcher that assigns Poisson arrivals to one of the servers. It has been shown in this setting that a class of load balancing policies including Join-the-Shortest-Queue (JSQ), Join-the-Idle-Queue (JIQ) [34] and variants of the Power-of-d-Choices (Pod) [37, 49] which sample a sufficiently large number of queues or exploit the parallelism of tasks within a job are able to achieve asymptotically zero waiting time for a sufficiently large N .

However, the above classical load balancing model may not be appropriate for certain modern cloud computing and data analytic applications due to the presence of job-server constraints. Under such constraints, a job can only be dispatched to a subset of the N servers. These constraints, often called locality constraints, are quite common in large-scale Machine Learning as a Service (MLaaS) and serverless computing services supported by cloud computing platforms (e.g., Microsoft Azure [36], Amazon Web Services [1], Google Cloud [22]). To give a concrete example, let us consider MLaaS. In this setting, various well-trained machine learning models are deployed on cloud platforms, say deep convolutional neural network (CNN) models for image classification and natural language processing (NLP) models. A user's image classification request can only be sent to the servers on which the CNN models have been loaded. As a result, it is not appropriate to assume that every request can be served by any server in the system. Other examples in which there are inherent job-server constraints include online video services, such as TikTok, Netflix and Youtube. In these applications, user requests can only be sent to servers with the required data (e.g., movies, music). The ultimate goal in all these modern applications is to achieve a fast response time and efficient resource (e.g., number of servers) usage while satisfying job-server constraints.

Inspired by these applications, in this paper, we take into account job-server constraints by considering a bipartite load balancing model. In this model, job-server constraints are abstracted by the edges in a bipartite graph, where the left nodes are called ports and the right nodes are called servers. In the model, each port represents a job of a particular type which requires a specific chunk of data or a specific machine learning model to execute, and thus can only be routed to specific servers. Each port ℓ corresponds to Poisson job arrivals with rate λ_ℓ . A job from a port ℓ can only be sent to server r such that (ℓ, r) is an edge of the graph. Jobs routed to a server r are queued in a buffer, and get service in a first-come first-serve (FCFS) manner. The service time of an arbitrary job at server r is exponentially distributed with rate μ_r (possibly different).

Our model is similar with that studied in [11] where JSQ is shown to be throughput optimal while no delay performance guarantee is provided. The bipartite graph model generalizes the load balancing model on graphs introduced in [8, 39]. In their model, jobs arrive at each node with a homogeneous rate, and an arbitrary job can be served by the node it arrives and its neighbors. It has been shown that in this setting JSQ achieves zero delays under certain assumptions on graph connectivity [39].

Inspired by the discussions above, we are particularly interested in the following question:

Are there simple policies that can achieve optimal response time in modern load balancing systems with both job-server constraints and service-rate heterogeneity?

1.1 Main Contribution

This paper affirmatively answers the above question by presenting optimal policies as well as performance bounds on the mean response time. The detailed contributions can be summarized as follows.

First, we consider two policies: Join-the-Fastest-of-the-Shortest-Queues (JFSQ), and Join-the-Fastest-of-the-Idle-Queues (JFIQ). We show that, under a 'well-connected' graph condition, they can asymptotically achieve the minimum response time in both the many-server regime (the system load $\lambda < 1$ is a constant while the number of servers $N \rightarrow \infty$) and sub Halfin-Whitt (HW) regime ($\lambda = 1 - N^{-\alpha}$ with $\alpha < 0.5$). The minimum response time metric is more stringent than the common

"zero queueing delays" discussed before, and is especially important in systems with heterogeneous servers. JFSQ and JFIQ are simple variants of JSQ and JIQ adapted to job-server constraints, but they break ties in JSQ and JIQ by choosing the fastest servers. Consequently, our results imply that JSQ and JIQ have asymptotic zero waiting time for homogeneous servers. They are practical since they only need comparisons between service speed rather than the exact service rates of servers. In addition to the asymptotic result, we also obtained finite-system bounds on the mean response time. Roughly speaking, we show that the difference between the mean response time in an N -server system and that in the limit is bounded by $O(\epsilon + ((1 - \lambda)\epsilon N)^{-1/2})$, where ϵ is a parameter related to the well-connectedness of the underlying bipartite graph, and λ reflects the load of the system. It is well known that Pod (power-of- d choices) routing does not achieve asymptotically zero waiting time while JSQ and JIQ do so in large parallel server systems. Our results show that JFIQ and JFSQ continue to achieve zero waiting time but in addition also provide mean response time optimality even in non-parallel server systems such as bipartite graphs. Additionally, the communication overhead between the router and the servers is very low under JFIQ.

Second, our theoretical results provide practical guidance in designing modern load balancing systems. Besides the two simple but efficient algorithms, the underlying 'well-connected' condition sheds light on the efficient deployment of various ML models or the required data among the servers. In particular, the key message is that each movie on Netflix or each ML model deployed on Microsoft Azure on average only needs to be loaded in $\omega(1)$ servers for nearly optimal mean response time. To give a concrete example, for a wide range of traffic patterns, we provide practical design guidelines showing that if movies or machine learning models are randomly loaded on servers according to some given probabilities, then JFSQ and JFIQ can guarantee nearly optimal mean response time with high probability. More precisely, Let L be the number of job types (e.g., the number of movies or machine learning models), and N be the number of servers. Our result indicates that on average, we only need $\omega\left(\frac{L+N}{(1-\lambda)^2}\right)$ copies of data for good delay performance. And if the arrival rates of jobs are uniform, then this number can be reduced to $\omega\left(\frac{L+N}{1-\lambda} \ln \frac{1}{1-\lambda}\right)$.

A key theoretical contribution of the paper is showing that a recently-developed Lyapunov drift method for studying parallel-server queueing systems can be generalized to bipartite graphs using two key ideas: (i) we demonstrate something akin to state-space collapse and resource pooling by exploiting the connectivity structure of the graph, and (ii) apply this idea iteratively twice, once to bound the number of jobs in fast servers that are busy in the large-system limit and a second time to bound the number of jobs in slow servers that the number is zero in the limit using a conditional geometric tail bound.

1.2 Related Work

There is a vast literature on efficient load balancing policies, mostly in the classical load balancing setting where there are N identical servers and the service rate is exponentially distributed. Upon arrival, an arbitrary job can be sent to any of the N servers. It is now well-known that in this setting JSQ is optimal [52] in a stochastic ordering sense. However, obtaining the exact steady state performance of JSQ is difficult. The problem is partly solved in [15] which establishes that the scaled queue length process of JSQ converges to a two-dimensional Ornstein-Uhlenbeck process, and the fraction of waiting jobs vanishes in the Halfin-Whitt heavy traffic regime. Although this result is on the process level, it is later confirmed for the steady state distribution by [6]. The tail of the distribution is further studied in [4].

Since JSQ has significant communication overhead in large-scale systems, alternative policies have been proposed and analyzed. One prominent policy is Power-of- d -Choices (Pod). In Pod, each arrival of jobs probes d random servers, and joins the one with the shortest queue. [40] first shows

that if $d \rightarrow \infty$, then both the fluid limit and the state occupancy distribution of Pod coincides with that of JSQ in many-server limit. It implies that Pod has zero waiting time in many-server limit. [40] also prove that the diffusion limit of Pod is the same as JSQ if $d = \omega(\sqrt{N} \log N)$ in the Halfin-Whitt heavy traffic regime, but it does not imply a steady-state performance guarantee. For the many-server regime, a line of works [17, 18] study the minimum required resources (such as memory, and communication overhead) to achieve zero waiting time.

When the system load λ can also approach 1 as N increases (i.e. many-server heavy-traffic regime), [31] shows that Pod can achieve asymptotic zero waiting time if $d = \omega(\frac{1}{1-\lambda})$ when $1 - \lambda = \omega(N^{-1/6})$. For a heavier-traffic regime, a recent breakthrough is the work [33]. In the sub Halfin-Whitt regime ($1 - \lambda = \omega(N^{-0.5})$), this work establishes asymptotic zero waiting property for a large class of policies including JSQ, JIQ and Pod with $d = O(\frac{\log N}{1-\lambda})$. The result is later extended to the Beyond-Halfin-Whitt regime ($1 - \lambda = \omega(N^{-1})$) [32], and to Coxian-2 service time distribution [30]. When $1 - \lambda = O(N^{-1})$, it is known that the waiting time must be positive for all load balancing policies [3, 25]. When jobs are divisible, [40, 53] shows similar result for Batch Sampling [41] and Batch-Filling [57], which are batch variants of Pod.

Proving optimality of load balancing algorithms is more complicated when servers are heterogeneous. Simple heuristics, nevertheless, have been proposed for decades. We note that a policy called *Never Queue* policy which is very similar to JFIQ was proposed in [43]. The *Never Queue* policy is analyzed in the case of a centralized queue, but not for load balancing systems. Many studies have focused on the heavy traffic regime where the system load converges to 1 while the number of servers is fixed. In this regime, JSQ was shown to be delay optimal by the drift method [14]. Later, [60] proves that a threshold policy is heavy-traffic optimal. The stability and optimality in heavy traffic of Pod for heterogeneous servers were studied recently by [29]. Moreover, [59] provides a simple criterion for load balancing algorithms to be heavy-traffic optimal. The assumption of heavy traffic can be relaxed to many-server heavy traffic regime when $1 - \lambda = o(N^{-4})$ [28, 58]. Nevertheless, the results mentioned above do not imply fast mean response time in the many-server regime, which is more practical for cloud platforms. For the many-server regime, work in [44] shows that JIQ has asymptotic zero waiting time as $N \rightarrow \infty$. However, this does not imply optimal mean response time since the service time of jobs varies in different servers. A recent work [19] takes heterogeneity into account by studying a system with fast and slow servers. Although [19] obtains mean-field limit for a variant policy of Pod, the result does not imply optimal mean response time.

Load balancing with job-server constraints is a long-standing problem in the literature. To the best of our knowledge, [16] was the first to study a model where different types of jobs can only served at certain sets of servers. Their model allows service rates to depend on both the job type and the server serving the job. The stability condition of a class of routing policies is examined when the service speed is independent of servers or independent of job types, and when there are two servers. When the system is not stochastic, [38] proposes an online load balancing algorithm with the optimal competitive ratio. A recent paper Cruise et al. [11] considers the stability of JSQ on the same model as ours while no delay guarantee is provided. In Cardinaels et al. [9], redundancy policies are explored in bipartite load balancing. They exploit a product-form steady state distribution in heavy traffic which implies heavy-traffic optimality. Besides these papers, there are also studies for load balancing on graphs. In [8, 21, 47], the impact of the graph structure on the performance of Pod is studied. Mukherjee et al. [39] utilizes a stochastic coupling method to prove that JSQ on graph can have the same performance as JSQ in the classical load balancing model in both the many-server regime and the Halfin-Whitt regime under certain graph constraints. Therefore, it implies that JSQ can also achieve zero waiting time in the many-server regime for a graph-based

model. However, the model in [39] only considers identical servers and homogeneous arrival rates of jobs, which is a special case of this paper.

We note that if servers share a central queue, then the bipartite graph model turns into the skill-based model studied in the call center literature [9, 20, 48]. It is shown in [20, 48] that the stationary distributions under a random assignment policy and several redundancy policies have product forms. One related result to us is that our model becomes the same as a skill-based model, and thus enjoys a product-form stationary distribution, if we send a job to a connected server with least amount of work in its buffer [9, 20]. Such policy is, however, impractical since workloads of jobs in cloud platforms suffer from volatility. Also, as [20] has pointed out, it is non-trivial to obtain bounds on mean response time just from the product-form results. A similar model with the skill-based model is investigated in [46], where there are multiple queues and servers, and a job to a queue can only be served by a server connected with the queue. Tsitsiklis and Xu [46] showed that in the many-server limit, even a sparse bipartite graph can have a large capacity region and a diminishing mean delay.

Our bipartite graph model also resembles other problems in the literature. One particular model is the job-server affinity model for data locality problems studied in [10, 51, 54, 55]. In the job-server affinity model, if an arbitrary job is served by a server with its data, it has a fast constant service rate. Otherwise, it has a slow service rate, meaning that this server has to fetch data from somewhere. However, the setting is not suitable in the context of MLaaS we discussed above. Here ML models are usually reconfigured on machines periodically, and a new request will only be routed to those servers with needed model [24]. Also, previous studies on job-server affinity models can only guarantee heavy-traffic delay optimality [51, 54, 55], which does not induce extremely fast mean response time required in cloud platforms.

From a methodological perspective, our paper builds on the drift method to obtain performance bounds. In this method, one exploits the fact that the steady-state expectation of suitable functions of the state of a Markov process does not change with time. This idea was developed in [14, 35, 50] for the heavy-traffic regime where the idea of using the tail bounds of [5, 27] to prove state-space collapse or resource pooling was introduced. The recent work in [33] developed a parallel approach for the many-server regime where they introduced the notion of generator coupling inspired by Stein's method in [7, 26, 45, 56] and designed a clever Lyapunov coupling to show that, for JSQ-type policies, the number of homogeneous servers utilized is large when the backlog is large. We will call this latter idea *state-space collapse* since it is similar to the notion of state-space collapse in the heavy-traffic regime. In this paper, we introduce new ideas to expand the applicability of the techniques [33] to networks of heterogeneous servers.

Contemporaneous to our work, in [42], the authors study the waiting time of JSQ(d) policies in bipartite graphs in the limit as the size of the graph goes to infinity. While the papers are motivated by related problems, the models and routing policies studied, and the results in the two papers are different. The authors in [42] consider the case of homogeneous servers with infinite buffers, and show that the performance of JSQ(d) in a bipartite graph with limited connectivity converges to the performance of the fully flexible system in terms of queue length (or waiting time) under appropriate connectivity conditions. In addition, they prove that the occupancy in steady state of the limited-connectivity system converges to the steady state of the fully flexible system. Our paper considers the case of heterogeneous arrival and service rates with finite buffers, and shows that the waiting time in the queue and blocking probability both go to zero in the large-system limit under the JFIQ and JFSQ routing policies. Additionally, the techniques used in the two papers are different. We use the drift method to obtain performance bounds for finite-sized systems while [42] uses process-level convergence techniques.

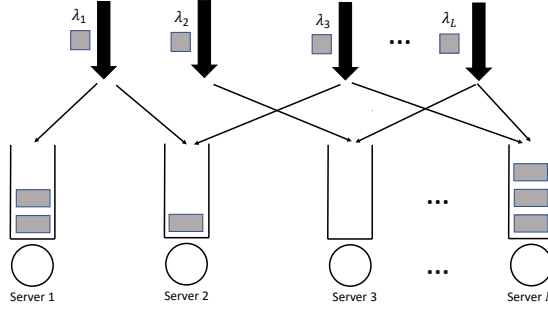


Fig. 1. An example of the bipartite graph model. In this instance, jobs from port 1 can only be routed to server 1 and server 2.

2 MODEL

We consider load balancing in a bipartite graph $G = (\mathcal{L}, \mathcal{R}, E)$ where \mathcal{L} and \mathcal{R} are the set of left nodes and right nodes, respectively, and E is the set of edges between these two sets of nodes. Nodes in \mathcal{L} are indexed as $\{1, 2, \dots, L\}$ with $L = |\mathcal{L}|$, and nodes in \mathcal{R} are indexed as $\{1, 2, \dots, N\}$ with $N = |\mathcal{R}|$. For a node $\ell \in \mathcal{L}$ (or $r \in \mathcal{R}$), define $\mathcal{N}_L(\ell)$ (or $\mathcal{N}_R(r)$) to be the set of right (or left) nodes it connects with. W.L.O.G., every $\mathcal{N}_L(\ell)$, $\mathcal{N}_R(r)$ is assumed to be non-empty. To distinguish between left and right nodes, we may refer to a node $\ell \in \mathcal{L}$ as port ℓ , and a node $r \in \mathcal{R}$ as server r . See Fig. 1 for an illustration.

Jobs arrive at port ℓ according to a Poisson process with rate λ_ℓ , and the goal is to route them to one of the servers connected to ℓ so as to minimize a certain performance metric of interest. It is assumed that every server has a finite buffer of size b . When a job is routed to a server that is currently processing another job, this new arrival will be placed in the buffer. But if there are already b jobs (including the one being served), the new arrival is blocked and lost forever. We assume that jobs in the buffer are served in a FCFS manner. The queue length Q_r of a server r is the number of jobs in the buffer plus one if there is a job running on the server.

To reflect the nature of server heterogeneity in a practical load balancing system, we assume that there are M types of servers. For a type m server, the service time of a job running on it is assumed to be exponentially distributed with mean $\frac{1}{\mu_m}$. The arrival processes to the ports and the service times of jobs are assumed to be independent. Denote the number of type m servers by N_m , and the type of a server r by t_r . Equivalently, we can write $N_m = N\alpha_m$ with $\alpha_m \in (0, 1)$, $\sum_{m=1}^M \alpha_m = 1$. We assume that there is sufficient service capacity, i.e., $\lambda_\Sigma = \sum_{\ell=1}^L \lambda_\ell < N \sum_{m=1}^M \mu_m \alpha_m$. W.L.O.G., server types are indexed in decreasing order of their service rates so that type one servers are the fastest, i.e., assume $\mu_1 > \mu_2 > \dots > \mu_M > 0$.

We study two routing policies, Join-the-Fastest-of-the-Shortest-Queues (JFSQ) and Join-the-Fastest-of-the-Idle-Queues (JFIQ) in bipartite load balancing systems. For JFSQ, upon the arrival of a job at port ℓ , we select a server r connected to port ℓ with the shortest queue length, that is, $r \in \arg \min_{r \in \mathcal{N}_L(\ell)} Q_r$. If there are multiple such servers, we select the one with the fastest service rate, i.e. largest μ_{t_r} , and break ties (if any) by randomly choosing one server. Alternatively, if we use JFIQ, we find an idle server $r \in \mathcal{N}_L(\ell)$ with the fastest service rate. If there is no idle server, we select one server from $\mathcal{N}_L(\ell)$ randomly. The question of interest in this paper is whether these two policies can achieve optimal job delays (at least for a large system) under appropriate conditions on the underlying bipartite graph. We note that our routing policies JFIQ and JFSQ reduce to JIQ

and JSQ, respectively, when all servers have the same service rates. In addition, the communication overhead of JFIQ in our system can be the same as that of JIQ in the classical load balancing system. That is, JFIQ incurs at most one amortized communication message per arrival. We will discuss the detailed implementation and associated complexity of JFSQ and JFIQ in the appendix.

2.1 State Representation

Before we proceed to state our results, we first state the notation that we will use in the paper. We use capital letters to denote random variables, such as $Q_r(t)$ for the queue length of server r at time t , and small letters to denote realizations.

Clearly, for the system considered in this paper, the sequence $\{Q(t) = (Q_1(t), \dots, Q_N(t))\}$ forms a Continuous Time Markov chain (CTMC). To ensure that there is a unique stationary distribution of $Q(t)$, we assume that the system starts at $Q(0) = \mathbf{0} = (0, \dots, 0)$, and the state space is restricted to the set of states reachable from state $\mathbf{0}$. Then since each queue is served in a FCFS manner, the Markov chain is irreducible, and hence has a unique stationary distribution because of a finite state space. For each state $\mathbf{q} = (q_1, \dots, q_N)$, let

$$s_{m,i}(\mathbf{q}) = \frac{1}{N} |\{r \in \mathcal{R}: q_r \geq i, t_r = m\}|$$

be the fraction of type m servers with queue length at least i . Besides, define

$$C_m(\mathbf{q}) = \sum_{i=1}^b s_{m,i}(\mathbf{q}),$$

which is the normalized (divided by N) number of jobs in type m servers.

Notation: As mentioned earlier, capital letters are reserved for random variables (such as $Q(t)$ for queue lengths at time t), and small letters are for realizations (such as \mathbf{q} for a queue-length state). We add a line on top of a variable meaning that it is in steady state (such as \bar{Q}). This paper makes use of asymptotic notations. For two positive functions $f(x), g(x)$, we write $f(x) = o(g(x))$ if $\sup \lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = 0$; write $f(x) = O(g(x))$ if $\sup \lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} < \infty$; write $f(x) = \Omega(g(x))$ if $\inf \lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} > 0$; write $f(x) = \omega(g(x))$ if $\inf \lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = \infty$. We summarize key notations of the paper in the Appendix.

3 MAIN RESULTS

We summarize our main results in this section. To be specific, our results provide an upper bound of the mean number jobs in the system under certain assumptions. This upper bound can directly imply asymptotic optimality of JFSQ and JFIQ in the sense of minimum mean response time, which we will define explicitly later. We also give a random graph construction of the graph G such that G can satisfy Assumption 2 with high probability.

3.1 Upper Bound of the Mean Number of Jobs

Let K be the minimum value such that $N \sum_{m=1}^K \mu_m \alpha_m > \lambda_\Sigma$. Such a K must exist by the assumption of sufficient service capacity. Assume that $\lambda_\Sigma = N \sum_{m=1}^K \mu_m \alpha_m (1 - \beta)$ where $0 < \beta \leq 1$, and denote $\lambda = \frac{\lambda_\Sigma}{N}$. Let

$$C_1^* = \alpha_1, \dots, C_{K-1}^* = \alpha_{K-1}, C_K^* = \frac{\lambda - \sum_{m=1}^{K-1} \mu_m \alpha_m}{\mu_K},$$

and let $C^* = \sum_{m=1}^K C_m^*$. Such definition is motivated by the mean-field limit of our system, which will be illustrated later. The following result provides lower bounds for the expected service time of an arbitrary job, and the mean number of jobs in the system.

PROPOSITION 3.1. *Suppose that the buffer size is infinite, i.e., $b = \infty$. Let \bar{Z} be the random variable denoting the service time of an arbitrary job. Then for any stable policy, the mean number of jobs in the system is lower bounded by NC^* , and*

$$\mathbb{E}[\bar{Z}] \geq \frac{C^*}{\lambda}. \quad (1)$$

The proof is provided in the appendix. Note that the assumption of an infinite buffer is to ensure no loss of arrivals due to buffers. Although this paper assumes finite buffers, our results prove that JFSQ and JFIQ have very low blocking probabilities. It means that our system under JFSQ and JFIQ is close to a system with infinite buffers. Therefore, Proposition 3.1 is consistent with our main results, and provides a lower bound on the mean response time for policies that can ensure nearly zero blocking probabilities.

For every $1 \leq m \leq K$, let \mathcal{R}_m be the set of servers of types 1 through m . Let $\hat{\beta} = \beta \sum_{m=1}^K \alpha_m$, and ϵ be a number in $(0, \frac{\hat{\beta}}{4}]$; we call ϵ the approximation error since we will later use this parameter to characterize the near optimality of our routing policies. For any subset $\mathcal{I} \subseteq \mathcal{R}$, define $\mathcal{N}_{\mathcal{R}}(\mathcal{I}) = \cup_{r \in \mathcal{I}} \mathcal{N}_{\mathcal{R}}(r)$ to be the set of ports connected to at least one server in \mathcal{I} , and $D_{\mathcal{I}} = \sum_{\ell \notin \mathcal{N}_{\mathcal{R}}(\mathcal{I})} \lambda_{\ell}$ be the sum of arrival rates at ports not connected to \mathcal{I} . Before stating our results on JFSQ and JFIQ, we first make a few assumptions on the system. Let $\tau_{1K} = \frac{\mu_1}{\mu_K}$, $\tau_{1M} = \frac{\mu_1}{\mu_M}$, $\tau_{KM} = \frac{\mu_K}{\mu_M}$. To reflect connectedness of the bipartite graph, we define $p_1 = \frac{\epsilon}{6b^2}$, $p_2 = \frac{\hat{\beta}}{2}$. In addition, let \tilde{d}_1, \tilde{d}_2 to be two constants that practitioners can control whose meaning would be clear in Assumption 2. Specifically, they can be any value such that $\tilde{d}_1 \leq \frac{\epsilon\mu_K}{12b^3}$, $\tilde{d}_2 \leq \frac{\epsilon\mu_K}{2b}$.

ASSUMPTION 1 (BUFFER SIZE). *For a fixed approximation parameter ϵ in $(0, \frac{\hat{\beta}}{4}]$, the buffer size b satisfies $6\sqrt{\tau_{1K}} \leq b \leq \left\lceil \left(\frac{\epsilon^2 N}{1152\tau_{1K} \ln N} \right)^{1/5} \right\rceil$.*

ASSUMPTION 2 (WELL CONNECTEDNESS). *The graph G satisfies the following conditions:*

- $D_{\mathcal{I}} \leq N\tilde{d}_1$ for any $\mathcal{I} \subseteq \mathcal{R}_{K-1}$ with $|\mathcal{I}| \geq Np_1$;
- $D_{\mathcal{I}} \leq N\tilde{d}_2$ for any $\mathcal{I} \subseteq \mathcal{R}_K$ with $|\mathcal{I}| \geq Np_2$.

In general, Assumption 2 requires that: 1) for jobs with only a few applicable servers, their arrival rates are small; 2) for jobs with large arrival rates, they are connected to many servers. Such a requirement enables that JFSQ and JFIQ behave almost the same as in a classical load balancing system even though there are additional job-server constraints. We are now ready to state the main result.

THEOREM 3.2. *Suppose that Assumptions 1 and 2 hold, and that the routing policy is either JFSQ or JFIQ. Then for a sufficiently large N , the following results hold:*

- (i) *the expected number of jobs in servers of the first K types divided by N is bounded as*

$$\mathbb{E} \left[\max \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) - (C^* + \epsilon), 0 \right) \right] \leq \frac{52\tau_{1K}b^2}{\epsilon N}; \quad (2)$$

- (ii) *if $K < M$, the expected number of jobs in the system divided by N is bounded as*

$$\mathbb{E} \left[\sum_{m=1}^M C_m(\bar{\mathbf{Q}}) \right] \leq C^* + \left(1 + \frac{\tau_{KM}}{2} \right) \epsilon + 2\sqrt{\frac{5\tau_{1M}b \ln N}{N}} + 60b^2 \sqrt{\frac{26\tau_{1K}\tau_{1M}}{\hat{\beta}\epsilon N}}; \quad (3)$$

(iii) the probability $p_{\mathcal{B}}$ that an arriving job is blocked is bounded as

$$p_{\mathcal{B}} \leq \frac{\tilde{d}_2}{\lambda} + \frac{52\tau_{1K}b^2}{\epsilon N}. \quad (4)$$

3.2 Asymptotic Optimality

Theorem 3.2 may be difficult to interpret since there are several parameters involved in the results. So let us interpret the result for an important special case which is perhaps the one that is practically most relevant. Suppose that the normalized arrival rate λ , the proportions of different types of servers $\{\alpha_m\}$, and ϵ are fixed. In most practical systems, the number of jobs that can wait at a server is small, so let us suppose that b is a fixed constant satisfying Assumption 2. Then, from (3), it is clear that the normalized expected number of jobs in the system is asymptotically equal to $C^* + O(\epsilon)$ in the many-server limit. The blocking probability goes to zero provided $\tilde{d}_2 = o(1)$ and the rate at which it goes to zero depends on rate at which \tilde{d}_2 decreases with N . From Proposition 3.1, the lower bound on the normalized number of jobs in an infinite buffer system is C^* . This suggests that JFSQ and JFIQ are near-optimal from the perspective of mean response time if the graph is reasonably well connected; we make this argument more general (by allowing many parameters to scale) and precise next.

To study the limit as N approaches infinity, we let $\{G_N = (\mathcal{L}_N, \mathcal{R}_N, E_N), N \geq 1\}$ be a sequence of bipartite graphs such that $|\mathcal{R}_N| = N$ and the buffer size of each server is given by b_N . Here, the number of servers, N , is allowed to scale, but the server-type distribution $(\alpha_1, \dots, \alpha_M)$, and the service rate of each type of servers, (μ_1, \dots, μ_M) , $\mu_1 > \dots > \mu_M$, are fixed. Further, the total arrival rates at ports in \mathcal{L}_N , λ_{Σ} , is assumed to be equal to $N \sum_{m=1}^K \mu_m \alpha_m (1 - \beta_N)$ for all G_N . As before, we can define a sequence of parameters $\{\epsilon_N, N \geq 1\}$ that quantify the approximation error where $\epsilon_N \in (0, \frac{\hat{\beta}_N}{4}]$, and $\hat{\beta}_N = \beta_N \sum_{m=1}^K \alpha_m$. Now we can discuss the asymptotic performance of a routing policy as $N \rightarrow \infty$.

Proposition 3.1 provides a lower bound on the expected service time of a job in the system with infinite buffers. We thus have the following definition of an (asymptotically) optimal routing policy in the bipartite load balancing system.

Definition 3.3 (Optimality in the Mean Response Time Sense). A stable routing policy is asymptotically optimal in the response time if the mean response time of jobs converges to $\frac{C^*}{\lambda}$ and the blocking probability goes to zero when $N \rightarrow \infty$.

We can see that optimality in the mean response time is a stronger metric than the common zero-waiting property discussed in the literature [17, 33, 44]. With this optimality, not only an arriving job has asymptotically zero waiting time, but it also has the minimum possible mean service time.

Then Theorem 3.2 immediately implies that both JFSQ and JFIQ are asymptotically optimal if the load of the system is moderate and the graph G_N is suitably well connected.

COROLLARY 3.4. *Suppose that ϵ_N is both $o(1)$ and $\omega(\ln(N)N^{-0.5})$, and that both Assumptions 1 and 2 hold for G_N when N is sufficiently large. Then as $N \rightarrow \infty$, both JFSQ and JFIQ are asymptotically optimal, and the expected queueing delay converges to zero for both policies.*

Due to the relationship between β_N and ϵ_N , it is not difficult to see that asymptotic optimality holds for arrival rates upto the sub-Halfin-Whitt regime. We refer the reader to the appendix for a proof of Corollary 3.4.

3.3 Random Graph Models

We now discuss when a bipartite graph can satisfy Assumption 2 in random graph models. Suppose the set of ports \mathcal{L} and the set of servers \mathcal{R} are fixed, but connections between them, i.e., the graph G , is not determined. This section considers a random graph G where port i connects with server j with probability z_{ij} . We devise an explicit construction of z_{ij} and show that such a random graph can satisfy Assumption 2 with a high probability. Our result first provides the construction of z_{ij} when ports can have different arrival rates. Later, by restricting the scope to homogeneous arrival rates among ports, we give a better construction where the graph G can have fewer edges. We are now ready to state our results.

THEOREM 3.5. *Let $H_j = \frac{2 \ln 2(N+L)/N}{p_j}$ for $j \in \{1, 2\}$. Consider the following construction of the graph G . For each port $\ell \in \mathcal{L}$ and each server $r \in \mathcal{R}$, the probability that they are connected is given by $z_{\ell,r} = \min\left(\frac{\lambda_\ell H_j}{N d_j}, 1\right)$ where $j = 1$ if $r \in \mathcal{R}_{K-1}$ and $j = 2$ if $r \in \mathcal{R}_K \setminus \mathcal{R}_{K-1}$. Then G satisfies Assumption 2 with probability at least $1 - 2^{-(N+L-1)}$. The expected total number of edges used in G_N scales as $O\left(\frac{(N+L)b^3}{\epsilon^2}\right)$.*

Next, we discuss the special case of homogeneous arrival rates.

THEOREM 3.6. *Suppose that all ports have the same arrival rates, that is, $\lambda_\ell \equiv \bar{\lambda}$ for all $\ell \in \mathcal{L}$. Then following the same construction of graph G in Theorem 3.5 but with $H_j = 6\left(-\ln p_j + \frac{\bar{d}_j}{p_j \bar{\lambda}} \ln \frac{2\mu_1}{\bar{d}_j}\right)$ for $j \in \{1, 2\}$, it holds that G satisfies Assumption 2 with probability at least $1 - 2\left(\frac{N}{N_{p_1}}\right)^{-1}$. The total number of edges in G_N scales as $O\left(\frac{(N+L)b^3}{\epsilon} \ln \frac{b}{\epsilon}\right)$.*

REMARK 1. *The previous two theorems indicate that to achieve asymptotically optimal mean response time and asymptotic zero waiting probability, the average number of connections of each port is only $O(\frac{1}{\epsilon^2})$ for heterogeneous arrival rates, and $O(\frac{1}{\epsilon} \ln \frac{1}{\epsilon})$ for homogeneous arrival rates, given that $L = \Omega(N)$, $b = O(1)$. When $1/(1 - \lambda) = O(1)$, we only require $\epsilon = o(1)$. Then the average number of edges connected to each port becomes $\omega(1)$. Therefore, for achieving very small loss probability and near-optimal response times, the number of edges in a random graph need to be only sparse compared to a fully connected graph.*

4 PROOF OF THE UPPER BOUND AND OPTIMALITY RESULTS

In this section, we provide the proofs of Theorem 3.2. These results respectively bound the mean number of jobs in a finite-size system and show the asymptotic optimality for JFSQ and JFIQ in the many-server limit and the sub Halfin-Whitt regime.

4.1 Proof Sketch

Ahead of the complete proof, we first provide a sketch of the proof reflecting intuitions behind it. Recall that the goal is to bound the mean number of jobs in the system divided by N , given by $\mathbb{E}\left[\sum_{m=1}^M C_m(\bar{\mathbf{Q}})\right]$. Here by definition, $C_m(\bar{\mathbf{Q}}) = \sum_{j=1}^b s_{m,j}(\bar{\mathbf{Q}})$. Our proof starts with the following observation about the mean-field limit for JFSQ and JFIQ in the heterogeneous system.

4.1.1 Mean-Field Limit. Ideally, if the load λ is a constant, then as $N \rightarrow \infty$, it holds that

$$s_{m,1}(\bar{\mathbf{Q}}) \approx \begin{cases} \alpha_m, & m < K \\ C_K^*, & m = K \\ 0, & m > K \end{cases} \quad \text{and} \quad s_{m,j}(\bar{\mathbf{Q}}) \approx 0, \quad \forall m = 1 \dots M, j = 2 \dots b. \quad (5)$$

Roughly speaking, this limit tells us that all the first $K - 1$ types of servers are busy, some servers of type K are busy, and all the servers with types greater than K are idle.

The intuition behind (5) is as follows. In the regimes we are considering, there are many servers. Therefore, a certain fraction of them must be idle. Then by the definition of JFIQ and JFSQ, all arrivals of jobs are routed to idle servers, at least in a fluid model. Therefore, the scaled number of waiting jobs (i.e., not in service), $\sum_{m=1}^M \sum_{j=2}^b S_{m,j}(\mathbf{Q})$ must converge to zero. For $S_{1,1}(\mathbf{Q}), \dots, S_{M,1}(\mathbf{Q})$, JFIQ and JFSQ always route jobs to fastest idle servers. Therefore, it must be the case that $s_{m,1}(\mathbf{Q})$ are filled from 1 to M until $\sum_{m=1}^M \mu_m s_{m,1}(\bar{\mathbf{Q}}) = \lambda$. That is to say, the total departure rate is equal to the total arrival rate. Therefore, we can ‘guess’ that the mean-field limit has the form (5).

Based on this limit, the scaled mean number of jobs can be decomposed as

$$\mathbb{E} \left[\sum_{m=1}^M C_m(\bar{\mathbf{Q}}) \right] = \mathbb{E} \left[\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right] + \mathbb{E} \left[\sum_{m=K+1}^M C_m(\bar{\mathbf{Q}}) \right]. \quad (6)$$

4.1.2 Lyapunov Drift Arguments. The drift argument starts by considering a Lyapunov function g and setting its drift in steady-state equal to zero. Since we are considering continuous-time Markov chains, this is equivalent to saying that $\mathbb{E} [Gg(\bar{\mathbf{Q}})] = 0$ where G is the generator of the Markov chain (defined explicitly later). Initially, let us focus on the total queue length in the first K types of servers (scaled by N) and thus, choose the Lyapunov function to be a function of the scaled total number of jobs in these servers and their queues, which we will call x . By an abuse of notation, we will rewrite the drift as $\mathbb{E} [Gg(x)] = 0$. However, this drift may be hard to analyze. Instead, suppose that the system was a simple deterministic fluid model of the form $\dot{x} = -\Delta$ for an appropriately $\Delta > 0$. The motivation for considering this fluid model is that, in the large-system limit, our system behaves like a single-server queue with simple fluid dynamics. If this fluid limit were the true system, then the drift of g becomes simply $-g'(x)\Delta$. We add and subtract this drift from the drift of the stochastic system to obtain $\mathbb{E} [Gg(x) - g'(x)\Delta + g'(x)\Delta] = 0$, which can be rewritten as

$$\mathbb{E} [g'(x)\Delta] = \mathbb{E} [Gg(x) - (-g'(x)\Delta)].$$

We are interested in getting a bound on the steady-state expectation of $h(x) = (x - C^* + \epsilon)^+$ where ϵ controls the approximation error. Therefore, we choose g such that $g'(x)\Delta = h(x)$ (this equality is sometimes called Stein’s equation). Thus, the drift equation becomes

$$\mathbb{E} [h(x)] = \mathbb{E} [Gg(x) - (-g'(x)\Delta)].$$

Now, it is easy to see that we can bound $\mathbb{E} [h(x)]$ if we can show that the drift of the Markov process $\mathbb{E} [Gg(x)]$ is approximately equal to $-g'(x)\Delta$. The rest of the proof involves studying $\mathbb{E} [Gg(x) - (-g'(x)\Delta)]$ by choosing $\Delta = \mu_1 \delta$ where $\delta > 0$. Define

$$W(\mathbf{q}) = \sum_{m=1}^K \mu_m s_{m,1}(\mathbf{q}),$$

which is the rate that a job completion will happen if we only consider servers of the first K types. In Lemma 4.3, we show that $\mathbb{E} [Gg(x) - (-g'(x)\Delta)]$ is approximately equal to

$$\frac{1}{\mu_1 \delta} \mathbb{E} \left[\mathbb{1} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \geq C^* + \epsilon + \frac{1}{N} \right\} h \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) (\lambda + \mu_1 \delta - W(\bar{\mathbf{Q}})) \right]. \quad (7)$$

We want to upper bound this expression by a quantity which is small when N is large. Note that $\sum_{m=1}^K C_m(\bar{\mathbf{Q}})$ is the total scaled queue length in the first K types of servers and $W(\bar{\mathbf{Q}}) = \sum_{m=1}^K \mu_m s_{m,1}(\bar{\mathbf{Q}})$ can be interpreted as the departure rate from these servers. Thus, the above

expression can be upper bounded by a small quantity if the following holds: whenever the total queue length is large, the departure rate exceeds the arrival rate with high probability.

To establish this fact, the mean-field limit (5) motivates us to show that $s_{m,1}(\bar{\mathbf{Q}}) \approx \alpha_m$ for $m < K$ and $s_{K,1}(\bar{\mathbf{Q}}) \approx C_K^*$. To be concrete, we show a two-stage state space collapse result through the following two Lyapunov functions (omitting extra technical terms):

$$\tilde{V}_1(\mathbf{q}) = \min \left(\sum_{m=1}^{K-1} \sum_{j=2}^b s_{m,j}(\mathbf{q}) + C_K(\mathbf{q}), \sum_{m=1}^{K-1} \alpha_m - \sum_{m=1}^{K-1} s_{m,1}(\mathbf{q}) \right) \quad (8)$$

$$\tilde{V}_2(\mathbf{q}) = \min \left(\sum_{m=1}^K \sum_{j=2}^b s_{m,j}(\mathbf{q}), \sum_{m=1}^{K-1} C_m^* + \tau_{1K}\delta - \sum_{m=1}^K s_{m,1}(\mathbf{q}) \right). \quad (9)$$

The well-connectedness condition in Assumption 2 and the routing policy (JFSQ and JFIQ) ensure that both of them have negative drifts when they are sufficiently large (Lemma 4.4 and Lemma 4.5). We now provide some intuition to explain how the well-connectedness condition plays a role in establishing the negative drift of these Lyapunov functions. We consider \tilde{V}_1 , the explanation for the other Lyapunov function is similar. If \tilde{V}_1 is large, it implies that both terms inside the min in (8) are large. In particular, by focusing on the second term, we note that a large \tilde{V}_1 implies that the (scaled) number of used servers $\sum_{m=1}^K s_{m,1}(\mathbf{q})$ is small. Equivalently, the number of idle servers is large. The well-connected condition simply states that the arrival rates to large subsets of servers is large. Thus, if \tilde{V}_1 is large, the number of empty servers is large which implies they have a large arrival rate, which in turn implies that the number of empty servers quickly decreases. The negative drift of \tilde{V}_1 and \tilde{V}_2 can be used to establish geometric tail bounds (Lemma 4.6) using standard drift arguments to show that they are small with high probability.

Observe that when $\sum_{m=1}^K C_m(\mathbf{q}) > C^* + \epsilon$, these two Lyapunov functions are both equal to the second term on their right hand side. Then in this case, $\sum_{m=1}^{K-1} s_{m,1}(\mathbf{q}) \approx \sum_{m=1}^{K-1} \alpha_m$, and $\sum_{m=1}^K s_{m,1}(\mathbf{q}) \approx \sum_{m=1}^K C_m^* + \tau_{1K}\delta$. It then implies $s_{K,1}(\mathbf{q}) \approx C_K^* + \tau_{1K}\delta$. Now that $\sum_{m=1}^K \mu_m C_m^* = \lambda$, it holds $W(\mathbf{q}) \approx \lambda + \mu_1\delta$ with high probability. We thus prove that (7) should be small, and it leads to a bound on the scaled mean number of jobs in the first K types of servers.

Now for the remaining types of servers, the mean-field limit (5) indicates that almost all of them are idle. We thus try to bound this third Lyapunov function, $\sum_{m=K+1}^M C_m(\bar{\mathbf{Q}})$. From the mean-field limit, we know that $\sum_{m=1}^K s_{m,1}(\bar{\mathbf{Q}}) \approx C^*$. Therefore, approximately $N \left(\sum_{m=1}^K \alpha_m - C^* \right)$ servers of the first K types are idle. Therefore, Assumption 2 ensures that very few jobs are routed to the remaining types of servers under JFSQ and JFIQ. By utilizing a conditional geometric tail bound (Lemma 4.6), we manage to show that $\sum_{m=K+1}^M C_m(\bar{\mathbf{Q}})$ is small with high probability, and finally obtain a bound on its mean.

For the complete proof of Theorem 3.2, since our theorem consists of three parts, we prove each of them in order, and combine them together at the end of this section.

4.2 Bound for the First K Types of Servers

The first result, which bounds the number of jobs in the first K types of servers, is the most important part in the theorem, which is restated as follows.

LEMMA 4.1. *Under Assumption 1 and Assumption 2, the expected number of jobs in servers of the first K types divided by N is bounded as*

$$\mathbb{E} \left[\max \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) - (C^* + \epsilon), 0 \right) \right] \leq \frac{52\tau_{1K}b^2}{\epsilon N} \quad (2)$$

if the routing policy is either JFSQ or JFIQ.

PROOF. Throughout this proof, we assume all assumptions in Lemma 4.1 are satisfied. Recall that the metric of interest is $\mathbb{E} \left[\max \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) - (C^* + \epsilon), 0 \right) \right]$, where $C^* = \sum_{m=1}^K C_m^*$. To simplify the notation, let $\eta = C^* + \epsilon$, and denote $h(x) = \max(x - \eta, 0)$. Our goal is thus to bound $\mathbb{E} \left[h(\sum_{m=1}^K C_m(\bar{\mathbf{Q}})) \right]$. The proof is motivated by the framework introduced in [33], and can be divided mainly into three parts, generator coupling, gradient bounds and state-space collapse.

Generator Coupling. We couple our system with a fluid model that is simple, but can well approximate the evolution of $h(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}))$. In particular, consider a fluid model $\dot{x} = -\mu_1\delta$ where $\delta = \frac{\mu_K}{6\mu_1 b^2}\epsilon$. Let $g(x)$ be the solution to the following Stein's equation of the fluid model,

$$\mu_1\delta g'(x) = h(x). \quad (10)$$

The solution is unique, and is given by

$$g(x) = \frac{\max(x - \eta, 0)^2}{2\mu_1\delta}, \quad g'(x) = \frac{\max(x - \eta, 0)}{\mu_1\delta}, \quad g''(x) = \begin{cases} 0, & x < \eta \\ \frac{1}{\mu_1\delta}, & x \geq \eta. \end{cases} \quad (11)$$

The next step is to couple our system with the fluid model through this Stein's equation.

To do so, recall that the system is a CTMC defined on queue lengths of servers, $\mathbf{Q}(t)$. Let G be the generator of our system such that for a queue state \mathbf{q} , and any function V defined on the state space,

$$GV(\mathbf{q}) = \sum_{\mathbf{q}'} r_{\mathbf{q},\mathbf{q}'} (V(\mathbf{q}') - V(\mathbf{q})) \quad (12)$$

where $r_{\mathbf{q},\mathbf{q}'}$ is the transition rate from state \mathbf{q} to state \mathbf{q}' . It is clear that $Gg(\mathbf{q})$ serves as an analog of the drift of function g at state \mathbf{q} in a discrete-time Markov chain as in [14]. To couple our system with the fluid model, we first need the following property, a key insight from [14] and [33].

LEMMA 4.2. *The expectation $\mathbb{E} \left[Gg(\sum_{m=1}^K C_m(\bar{\mathbf{Q}})) \right]$ is equal to 0.*

Then the two systems can be coupled by seeing that

$$\mathbb{E} \left[h \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) \right] = \mathbb{E} \left[g' \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) (\mu_1\delta) \right] \quad (13)$$

$$= \mathbb{E} \left[Gg \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) - g' \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) (-\mu_1\delta) \right]. \quad (14)$$

As a result, to bound $\mathbb{E} \left[h \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) \right]$, it is equivalent to bound (14).

Gradient Bounds. We now utilize the explicit form of $g(x)$ in (11) to bound (14). First by definition, it holds that for a state \mathbf{q} ,

$$\begin{aligned} Gg\left(\sum_{m=1}^K C_m(\mathbf{q})\right) &= \sum_{\mathbf{q}'} r_{\mathbf{q},\mathbf{q}'} \left(g\left(\sum_{m=1}^K C_m(\mathbf{q}')\right) - g\left(\sum_{m=1}^K C_m(\mathbf{q})\right) \right) \\ &= \lambda_{\Sigma} (1 - P_k(\mathbf{q})) \left(g\left(\sum_{m=1}^K C_m(\mathbf{q}) + \frac{1}{N}\right) - g\left(\sum_{m=1}^K C_m(\mathbf{q})\right) \right) \quad (\text{Arrival transitions}) \end{aligned} \quad (15)$$

$$+ NW(\mathbf{q}) \left(g\left(\sum_{m=1}^K C_m(\mathbf{q}) - \frac{1}{N}\right) - g\left(\sum_{m=1}^K C_m(\mathbf{q})\right) \right) \quad (\text{Departure transitions}) \quad (16)$$

where $P_k(\mathbf{q})$ is the probability that an arriving job is routed to a server of type greater than K , and $W(\mathbf{q}) = \sum_{m=1}^K \mu_m s_{m,1}(\mathbf{q})$. Then by (14), we can get

$$\mathbb{E} \left[h\left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}})\right) \right] \leq \mathbb{E} \left[g'\left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}})\right) (\mu_1 \delta) \right] \quad (17)$$

$$+ \lambda_{\Sigma} \left(g\left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) + \frac{1}{N}\right) - g\left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}})\right) \right) \quad (18)$$

$$+ NW(\bar{\mathbf{Q}}) \left(g\left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) - \frac{1}{N}\right) - g\left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}})\right) \right) \quad (19)$$

where we omit the term $P_k(\bar{\mathbf{Q}})$ from (16) since $g(x)$ is an increasing function by (11). Now to simplify the equation, we can do Taylor's expansion on (18) and (19), and apply gradient bounds of $g(x)$. The result is summarized as follows whose proof is provided in the appendix.

LEMMA 4.3. *It holds that*

$$\mathbb{E} \left[h\left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}})\right) \right] \leq \mathbb{E} \left[\mathbb{1} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \geq \eta + \frac{1}{N} \right\} g'\left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}})\right) (\mu_1 \delta + \lambda - W(\bar{\mathbf{Q}})) \right] + \frac{38b^2 \tau_{1K}}{\epsilon N}. \quad (20)$$

The remaining step is to bound the first term on the right hand side in (20), which is the main part of this proof. The key insight is that as long as $W(\mathbf{q}) \geq \lambda + \mu_1 \delta$, it holds that the contribution of \mathbf{q} to the first term would be at most zero. Furthermore, this property only needs to hold when $\sum_{m=1}^K C_m(\mathbf{q}) \geq \eta + \frac{1}{N}$ due to the indicator function. To justify this result, we establish two state space collapse results as follows.

State Space Collapse. Recall that $\sum_{m=1}^K C_m(\mathbf{q})$ is the number of jobs in servers of the first K types divided by N . The intuition is to show that when this number is large, it holds that with high probability,

$$s_{1,1}(\mathbf{q}) = C_1^*, \dots, s_{K-1,1}(\mathbf{q}) = C_{K-1}^*, s_{K,1} > C_K^*. \quad (21)$$

That is to say, almost all servers of the first $K-1$ types are busy. And enough type- K servers are busy such that their total departure rates (or works produced by these servers) are sufficient for the total arrival rate λ_{Σ} .

The following lemma indirectly shows that unless $\sum_{m=1}^K C_m(\mathbf{q})$ is small, $\sum_{m=1}^K s_{m,1}(\mathbf{q}) \approx \sum_{m=1}^{K-1} \alpha_m$. In particular, it designs a Lyapunov function closely related to the above property. Due to space limitations, the proof is deferred to the appendix.

LEMMA 4.4. *Consider the following Lyapunov function*

$$V_1(\mathbf{q}) = \min \left(\sum_{j=1}^b s_{K,j}(\mathbf{q}) + \sum_{m=1}^{K-1} \sum_{j=2}^b s_{m,j}(\mathbf{q}), \sum_{m=1}^{K-1} C_m^* - \sum_{m=1}^{K-1} s_{m,1}(\mathbf{q}) \right). \quad (22)$$

It holds that if $V_1(\mathbf{q}) \geq B_1 := \tau_{1K}\delta$, then $GV_1(\mathbf{q}) \leq \frac{-\mu_1\delta}{2b}$.

In addition to Lemma 4.4 that focuses on the first $K-1$ types of servers, the following lemma provides another Lyapunov function. This function is later used together with Lemma 4.4 to show that if $\sum_{m=1}^K C_m(\mathbf{q})$ is large, then a certain number of type K servers are busy. It then complements the goal in (21). The proof of this lemma is similar to that of Lemma 4.4, and is provided in the appendix.

LEMMA 4.5. *Consider the following Lyapunov function*

$$V_2(\mathbf{q}) = \min \left(\sum_{m=1}^K \sum_{j=2}^b s_{m,j}(\mathbf{q}), \sum_{m=1}^K C_m^* + B_2 + 3\tau_{1K}\bar{\delta} - \sum_{m=1}^K s_{m,1}(\mathbf{q}) \right) \quad (23)$$

where $\bar{\delta} := \tau_{1K}\delta$, and $B_2 := \frac{1}{2}\epsilon + \bar{\delta}$. It holds that if $V_2(\mathbf{q}) \geq B_2$, then $GV_2(\mathbf{q}) \leq -\frac{\mu_1\delta}{b}$.

To apply the above two lemmas, we need the following geometric tail bound from [53], which originates in [5, 50]. This lemma translates the fact that a Lyapunov function has a negative drift to the property that the function is within a certain region with high probability.

LEMMA 4.6. *Consider a continuous time Markov chain $\{\mathbf{S}(t) : t \geq 0\}$ on a finite state space \mathcal{S} . Assume that it has a unique stationary distribution. For a Lyapunov function $V : \mathcal{S} \rightarrow [0, +\infty)$, define $GV(\mathbf{s}) = \sum_{\mathbf{s}' \in \mathcal{S}} r_{\mathbf{s},\mathbf{s}'}(V(\mathbf{s}') - V(\mathbf{s}))$ where $r_{\mathbf{s},\mathbf{s}'}$ is the transition rate from state \mathbf{s} to \mathbf{s}' .*

Suppose that

$$v_{\max} := \sup_{\mathbf{s}, \mathbf{s}' \in \mathcal{S} : r_{\mathbf{s},\mathbf{s}'} > 0} |V(\mathbf{s}) - V(\mathbf{s}')| < \infty; \quad f_{\max} := \max \left\{ 0, \sup_{\mathbf{s} \in \mathcal{S}} \sum_{\mathbf{s}' : V(\mathbf{s}') > V(\mathbf{s})} r_{\mathbf{s},\mathbf{s}'} (V(\mathbf{s}') - V(\mathbf{s})) \right\} < \infty.$$

Given a set \mathcal{E} . If for some $B > 0, \gamma > 0, \xi \geq 0$, it holds: 1) $GV(\mathbf{s}) \leq -\gamma$ when $V(\mathbf{s}) \geq B$ and $\mathbf{s} \in \mathcal{E}$; 2) $GV(\mathbf{s}) \leq \xi$ when $V(\mathbf{s}) \geq B$ and $\mathbf{s} \notin \mathcal{E}$,

then for every positive integer j , if $\bar{\mathbf{S}}$ is the steady-state random variable, it holds

$$\mathbb{P} \{V(\bar{\mathbf{S}}) \geq B + 2v_{\max}j\} \leq \left(\frac{f_{\max}}{f_{\max} + \gamma} \right)^j + \left(\frac{\xi}{\gamma} + 1 \right) \mathbb{P} \{\mathbf{s} \notin \mathcal{E}\}. \quad (24)$$

Based on Lemma 4.6, we can bound the probability that $V_1(\mathbf{q})$ or $V_2(\mathbf{q})$ is large in the following result.

LEMMA 4.7. *Let $\chi = 96\tau_{1K}b^3 \ln N$. With the same notation in Lemma 4.4 and Lemma 4.5, it holds that*

$$\mathbb{P} \left\{ V_1(\bar{\mathbf{Q}}) \geq B_1 + \frac{\chi}{\epsilon N} \right\} \leq N^{-2}; \quad \mathbb{P} \left\{ V_2(\bar{\mathbf{Q}}) \geq B_2 + \frac{\chi}{\epsilon N} \right\} \leq N^{-2}. \quad (25)$$

PROOF. Note that under the notation in Lemma 4.6, we have for both $V_1(\mathbf{q})$ and $V_2(\mathbf{q})$, $v_{\max} = \frac{1}{N}$, and $f_{\max} \leq \mu_1$. We first bound $\mathbb{P} \{V_1(\mathbf{q}) \geq B_1 + \frac{\chi}{\epsilon N}\}$. Since by Lemma 4.4, when $V_1(\mathbf{q}) \geq B_1$, it holds

$GV_1(\mathbf{q}) \leq \frac{-\mu_1 \delta}{2b}$. Then by taking the set \mathcal{E} to be the empty set and taking $j_1 = \frac{8b}{\delta} \log N$, Lemma 4.6 shows that

$$\mathbb{P}\{V_1(\mathbf{q}) \geq B_1 + 2v_{\max} j_1\} \leq \left(1 + \frac{\delta}{2b}\right)^{-j_1} \leq \exp\left(-\frac{j_1 \delta}{4b}\right) = N^{-2} \quad (26)$$

where the last inequality comes from the fact that $\ln(1+x) \geq x/2$ for $x \in [0, 1]$. We can easily verify that $2v_{\max} j_1 = \frac{2}{N} \cdot \frac{48\mu_1 b^3}{\mu_K \epsilon} = \frac{\chi}{\epsilon N}$. Similarly, take $j_2 = \frac{4b}{\delta} \log N$ for $V_2(\mathbf{q})$. Together with Lemma 4.5, Lemma 4.6 shows that

$$\mathbb{P}\{V_2(\mathbf{q}) \geq B_2 + 2v_{\max} j_2\} \leq \left(1 + \frac{\delta}{b}\right)^{-j_2} \leq \exp\left(-\frac{j_2 \delta}{2b}\right) = N^{-2}. \quad (27)$$

We complete the proof by noticing that $2v_{\max} j_2 = \frac{2}{N} \cdot \frac{24\mu_1 b^3}{\mu_K \epsilon} \leq \frac{\chi}{\epsilon N}$. \square

Completing the Whole Proof. Finally, combining Lemma 4.7 with Lemma 4.3 help us complete the proof. To see why, recall that it remains to bound

$$\mathbb{E} \left[\mathbb{1} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \geq \eta + \frac{1}{N} \right\} g' \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) (\lambda + \mu_1 \delta - W(\bar{\mathbf{Q}})) \right]. \quad (28)$$

Let event $\mathcal{D} = \{V_1(\bar{\mathbf{Q}}) \leq B_1 + \frac{\chi}{\epsilon N}\} \cap \{V_2(\bar{\mathbf{Q}}) \leq B_2 + \frac{\chi}{\epsilon N}\}$. It holds that

$$\begin{aligned} (28) &\leq \mathbb{E} \left[\mathbb{1} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \geq \eta + \frac{1}{N} \right\} g' \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) (\lambda + \mu_1 \delta - W(\bar{\mathbf{Q}})) \middle| \mathcal{D} \right] + g'(b) \mu_1 (1 + \delta) \mathbb{P}\{\bar{\mathcal{D}}\} \\ &\leq \mathbb{E} \left[\mathbb{1} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \geq \eta + \frac{1}{N} \right\} g' \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) (\lambda + \mu_1 \delta - W(\bar{\mathbf{Q}})) \middle| \mathcal{D} \right] + \frac{2b}{\delta N^2} (1 + \delta) \end{aligned} \quad (29)$$

where the first inequality is by the law of total probability and the fact that $g'(x)$ is a positive increasing function, that $\sum_{m=1}^K C_m(\mathbf{q}) \leq b$ for all possible \mathbf{q} , and that $\lambda \leq \mu_1$, and the second inequality is by Lemma 4.7 that shows $\mathbb{P}\{\bar{\mathcal{D}}\} \leq \frac{2}{N^2}$.

Therefore, it is sufficient to bound the first term in (29). The following lemma shows that this term is indeed non-positive.

LEMMA 4.8. *For any \mathbf{q} such that $V_1(\mathbf{q}) \leq B_1 + \frac{\chi}{\epsilon N}$ and $V_2(\mathbf{q}) \leq B_2 + \frac{\chi}{\epsilon N}$, it holds that*

$$\mathbb{1} \left\{ \sum_{m=1}^K C_m(\mathbf{q}) \geq \eta + \frac{1}{N} \right\} (\lambda + \mu_1 \delta - W(\mathbf{q})) \leq 0. \quad (30)$$

PROOF. W.L.O.G., we can directly assume $\sum_{m=1}^K C_m(\mathbf{q}) \geq \eta + \frac{1}{N}$. Otherwise, (30) is already zero. Then the key step is to show $W(\mathbf{q}) = \sum_{m=1}^K \mu_m s_{m,1}(\mathbf{q}) \geq \lambda + \mu_1 \delta$. By the definition of $V_1(\mathbf{q})$ in (23), since $\sum_{m=1}^K C_m(\mathbf{q}) \geq \eta + \frac{1}{N}$, it holds that $V_1(\mathbf{q}) = \sum_{m=1}^{K-1} C_m^* - \sum_{m=1}^{K-1} s_{m,1}(\mathbf{q})$. Furthermore, as $V_1(\mathbf{q}) \leq B_1 + \frac{\chi}{\epsilon N}$ and $C_m^* = \alpha_m$ for $m < K$, it holds that

$$\sum_{m=1}^{K-1} s_{i,1}(\mathbf{q}) \geq \sum_{m=1}^{K-1} \alpha_m - \left(B_1 + \frac{\chi}{\epsilon N}\right). \quad (31)$$

Since $s_{m,1}(\mathbf{q}) \leq \alpha_m$ for all m , the total departure rate of servers of the first $K-1$ types is at least

$$\sum_{m=1}^{K-1} \mu_m s_{m,1}(\mathbf{q}) \geq \sum_{m=1}^{K-1} \mu_m \alpha_m - \mu_1 \left(B_1 + \frac{\chi}{\epsilon N}\right). \quad (32)$$

Then for $s_{K,1}(\mathbf{q})$, recall the definition of $V_2(\mathbf{q})$ in (22). To show that $V_2(\mathbf{q})$ is equal to the second term in its definition, note that

$$B_2 + 3\tau_{1K}\bar{\delta} = \frac{1}{2}\epsilon + \tau_{1K}\delta + 3\tau_{1K}^2\delta \leq \frac{1}{2} + \frac{2\tau_{1K}\epsilon}{3b^2} \leq \epsilon.$$

Then since $\sum_{m=1}^K C_m(\mathbf{q}) \geq \sum_{m=1}^K C_m^* + \epsilon + \frac{1}{N}$, it holds $\sum_{m=1}^K C_m(\mathbf{q}) \geq \sum_{m=1}^K C_m^* + B_2 + 3\tau_{1K}\bar{\delta}$. Therefore, $V_2(\mathbf{q})$ is equal to $\sum_{m=1}^K C_m^* + B_2 + 3\tau_{1K}\bar{\delta} - \sum_{m=1}^K s_{m,1}(\mathbf{q})$, the second term in (22). By assumption, $V_2(\mathbf{q}) \leq B_2 + \frac{\chi}{\epsilon N}$. As a result,

$$\sum_{m=1}^K s_{m,1}(\mathbf{q}) \geq \sum_{m=1}^K C_m^* + 3\tau_{1K}\bar{\delta} - \frac{\chi}{\epsilon N}, \quad (33)$$

and

$$s_{K,1}(\mathbf{q}) \geq C_K^* + 3\tau_{1K}\bar{\delta} - \frac{\chi}{\epsilon N} \quad (34)$$

because $s_{m,1}(\mathbf{q}) \leq \alpha_m = C_m^*$ for $m < K$. From (32) and (34), it holds

$$W(\mathbf{q}) = \sum_{m=1}^{K-1} \mu_m s_{m,1}(\mathbf{q}) + \mu_K s_{K,1}(\mathbf{q}) \geq \sum_{m=1}^{K-1} \mu_m \alpha_m + \mu_K C_K^* + 3\mu_K \tau_{1K} \bar{\delta} - \mu_1 B_1 - 2\frac{\mu_1 \chi}{\epsilon N} \quad (35)$$

$$\geq \lambda + 2\frac{\mu_1^2}{\mu_K} \delta - \frac{192\mu_1^2 b^3}{\mu_K \epsilon N} \ln(N) \geq \lambda + \mu_1 \delta \quad (36)$$

where the last inequality is because $\mu_1 > \mu_K$, and $\frac{\mu_1^2}{\mu_K} \delta \geq \frac{192\mu_1^2 \ln(N)}{\mu_K \epsilon N} b^3$ by Assumption 1. The inequality (36) immediately implies the desired result. \square

To conclude the proof of Lemma 4.1, by Lemma 4.3, the bound in (29) and Lemma 4.8, it holds

$$\mathbb{E} \left[h \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) \right] \leq \frac{2b}{\delta N^2} (1 + \delta) + \frac{38b^2 \tau_{1K}}{\epsilon N} \leq \frac{12b^3 \tau_{1K}}{\epsilon N^2} + \frac{2b}{N^2} + \frac{38b^2 \tau_{1K}}{\epsilon N} \leq \frac{52b^2 \tau_{1K}}{\epsilon N}. \quad (37)$$

\square

4.3 Bound for the Remaining Servers

Since Lemma 4.1 only bounds the mean number of jobs in servers of the first K types, we need the following result for the remaining servers in the system. This result shows that very few jobs will be served by servers of the last $M - K$ types. Note that if $K = M$, then Lemma 4.1 already bounds the mean number of jobs in the system.

LEMMA 4.9. *Suppose $K < M$. Under Assumption 1 and Assumption 2, if N is sufficiently large, the expected number of jobs in servers of the last $M - K$ types divided by N is bounded as*

$$\mathbb{E} \left[\sum_{m=K+1}^M C_m(\bar{\mathbf{Q}}) \right] \leq \frac{\tilde{d}_2 b}{\mu_M} + 2\sqrt{\frac{5\tau_{1M} b \ln N}{N}} + 8b^2 \sqrt{\frac{26\tau_{1K} \tau_{1M}}{\hat{\beta} \epsilon N}}. \quad (38)$$

if the routing policy is either JFSQ or JFIQ.

PROOF. To prove this result, let us consider the Lyapunov function $V_3(\mathbf{q}) = \sum_{m=K+1}^M C_m(\mathbf{q})$. Then by showing that this function has a negative drift when outside of a region, we can obtain a bound on its expectation. To do so, define B_3 as

$$B_3 = \frac{1}{\mu_M} \left(\tilde{d}_2 b + \sqrt{\mu_1 \mu_M \left(\frac{5b \ln(N)}{N} + \frac{416\tau_{1K} b^4}{\hat{\beta} \epsilon N} \right)} \right). \quad (39)$$

Let $\mathcal{E}_K = \{\mathbf{q}: \sum_{m=1}^K C_m(\mathbf{q}) \leq C^* + \frac{\hat{\beta}}{2}\}$. It holds that $\bar{\mathbf{Q}}$ lies in \mathcal{E}_K with high probability by the following lemma whose proof is in the appendix.

LEMMA 4.10. *For any $\Delta \geq \frac{\hat{\beta}}{2}$, it holds $\mathbb{P}\{\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) > C^* + \Delta\} \leq \frac{104\tau_{1K}b^2}{\Delta\epsilon N}$.*

By Lemma 4.10, it holds that $\mathbb{P}\{\bar{\mathbf{Q}} \notin \mathcal{E}_K\} \leq \frac{208\tau_{1K}b^2}{\hat{\beta}\epsilon N}$. Then it is natural to discuss the drift of $V_3(\mathbf{q})$ when it is greater than B_3 by conditioning on whether \mathbf{q} is in \mathcal{E}_K or not. The result is summarized in this lemma, and the proof is in the appendix.

LEMMA 4.11. *When $V_3(\mathbf{q}) \geq B_3$, it holds that*

- if $\mathbf{q} \in \mathcal{E}_K$, the drift is bounded as $GV_3(\mathbf{q}) \leq -\frac{B_3\mu_M}{b} + \tilde{d}_2$;
- if $\mathbf{q} \notin \mathcal{E}_K$, the drift is bounded as $GV_3(\mathbf{q}) \leq \mu_1$.

We now apply Lemma 4.6. Under the notation of that lemma, it holds $v_{\max} = \frac{1}{N}$, $f_{\max} \leq \mu_1$ for $V_3(\mathbf{q})$. Let $\gamma := \frac{B_3\mu_M}{b} - \tilde{d}_2$, and take $j_3 = \frac{2\mu_1 \ln(N)}{\gamma}$. Applying Lemma 4.6 and using Lemma 4.11, it holds that

$$\mathbb{P}\left\{V_3(\bar{\mathbf{Q}}) > B_3 + \frac{2j_3}{N}\right\} \leq \left(1 + \frac{\gamma}{\mu_1}\right)^{-j_3} + \left(\frac{\mu_1}{\gamma} + 1\right) \mathbb{P}\{\mathbf{q} \notin \mathcal{E}_K\} \leq N^{-2} + \frac{416\mu_1\tau_{1K}b^2}{\beta\epsilon N} \quad (40)$$

where the last inequality is because $\gamma < \mu_1$ when N is sufficiently large. Furthermore, the expectation of $V_3(\bar{\mathbf{Q}})$ can be bounded as

$$\mathbb{E}[V_3(\bar{\mathbf{Q}})] \leq \mathbb{E}\left[V_3(\bar{\mathbf{Q}}) \mid V_3(\bar{\mathbf{Q}}) \leq B_3 + \frac{2j_3}{N}\right] + \mathbb{E}\left[V_3(\bar{\mathbf{Q}}) \mid V_3(\bar{\mathbf{Q}}) > B_3 + \frac{2j_3}{N}\right] \mathbb{P}\left\{V_3(\bar{\mathbf{Q}}) > B_3 + \frac{2j_3}{N}\right\} \quad (41)$$

$$\leq B_3 + \frac{4\mu_1 \ln(N)}{\gamma N} + b \left(N^{-2} + \frac{416\mu_1\tau_{1K}b^2}{\beta\epsilon N}\right) \quad (42)$$

$$\leq B_3 + \frac{5\mu_1 \ln(N)}{\gamma N} + \frac{416\mu_1\tau_{1K}b^3}{\hat{\beta}\epsilon\gamma N}. \quad (43)$$

The definition of B_3 in (39) and that of γ immediately give the desired result. \square

4.4 Throughput Guarantee and the Proof of Theorem 3.2

The next lemma provides a bound on the blocking probability, and thus characterizes the effective throughput of the system. Due to space limitations, the reader is referred to the appendix for the proof.

LEMMA 4.12. *Under Assumptions 1 and 2, the probability p_B that an arriving job is blocked is bounded as*

$$p_B \leq \frac{\tilde{d}_2}{\lambda} + \frac{52\tau_{1K}b^2}{\epsilon N}. \quad (4)$$

Wrapping up above lemmas, we can conclude the proof of Theorem 3.2.

PROOF OF THEOREM 3.2. The first result and third result in Theorem 3.2 corresponds to Lemma 4.1 and 4.12. For the second result, notice that Lemma 4.1 implies

$$\mathbb{E}\left[\sum_{m=1}^K C_m(\bar{\mathbf{Q}})\right] \leq C^* + \epsilon + \frac{52\tau_{1K}b^2}{\epsilon N}. \quad (44)$$

Then combining (44) and (4) in Lemma 4.9 and the assumption that $\tilde{d}_2 \leq \frac{\epsilon \mu_K}{2b}$ in Assumption 2, it holds

$$\begin{aligned} \mathbb{E} \left[\sum_{m=1}^M C_m(\bar{\mathbf{Q}}) \right] &= \mathbb{E} \left[\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right] + \mathbb{E} \left[\sum_{m=K+1}^M C_m(\bar{\mathbf{Q}}) \right] \\ &\leq C^* + \epsilon + \frac{52\tau_{1K}b^2}{\epsilon N} + \frac{\tilde{d}_2 b}{\mu_M} + 2\sqrt{\frac{5\tau_{1M}b \ln N}{N}} + 8b^2 \sqrt{\frac{26\tau_{1K}\tau_{1M}}{\hat{\beta}\epsilon N}} \\ &\leq C^* + \left(1 + \frac{\mu_K}{2\mu_M}\right)\epsilon + 2\sqrt{\frac{5\tau_{1M}b \ln N}{N}} + 60b^2 \sqrt{\frac{26\tau_{1K}\tau_{1M}}{\hat{\beta}\epsilon N}}, \end{aligned}$$

which is exactly (3). \square

5 PROOF OF THE RANDOM GRAPH RESULTS

In this section, we prove Theorem 3.5. Since similar proof holds for Theorem 3.6, we provide that proof in the appendix.

Proof Sketch. The result is proved by showing that almost every pair of large enough subsets of \mathcal{L}, \mathcal{R} shares edges between the two sets because of the random graph structure. To show this fact, we first bound the probability that two given subsets are disconnected. Then the union bound concludes the proof since the total number of pairs of subsets is given by 2^{L+N} .

5.1 Proof of Theorem 3.5

PROOF. Recall the definition of $p_1, p_2, \tilde{d}_1, \tilde{d}_2$ in Assumption 2. W.L.O.G., assume Np_j is an integer for $j = 1, 2$. Otherwise, we can raise p_j to satisfy this condition since the size of a subset must be an integer. Suppose that we generate a bipartite graph G as in Theorem 3.5. Let C_j be the event that G violates the j -th condition in Assumption 2. We bound $\mathbb{P}\{C_j\}$ separately. To simplify the notation, let us denote $\mathcal{R}^1 = \mathcal{R}_{K-1}, \mathcal{R}^2 = \mathcal{R}_K$. And let us write $z_{\ell,r}$ be the probability that a port ℓ connects with a server r in the graph G .

First, define $\mathcal{D}_{\mathcal{K},I}$ as the event that a subset \mathcal{K} of \mathcal{L} has no edges with a subset I of \mathcal{R} . Then for $j = 1, 2$,

$$C_j = \bigcup_{\substack{\mathcal{K} \subseteq \mathcal{L}: \sum_{\ell \in \mathcal{K}} \lambda_\ell > N\tilde{d}_j \\ I \subseteq \mathcal{R}^j: |I| \geq Np_j}} \mathcal{D}_{\mathcal{K},I}. \quad (45)$$

Fix $j \in \{1, 2\}$. Let \mathcal{K} be any subset of \mathcal{L} satisfying $\sum_{\ell \in \mathcal{K}} \lambda_\ell > N\tilde{d}_j$, and I be any subset of \mathcal{R}^j satisfying $|I| \geq Np_j$. We want to bound $\mathbb{P}\{\mathcal{D}_{\mathcal{K},I}\}$. Notice that by Assumption 2, it holds $p_1 < p_2, \tilde{d}_1 < \tilde{d}_2$, and $\frac{\tilde{d}_2}{H_2} \geq \frac{\tilde{d}_1}{H_1}$. Then by the construction of G , if there is a port ℓ in \mathcal{K} such that $\lambda_\ell \geq N\tilde{d}_j H_j$, this port must be connected to all servers in \mathcal{R}^j , meaning that $\mathbb{P}\{\mathcal{D}_{\mathcal{K},I}\} = 0$. Therefore, we can assume that such port does not exist. Recall that $z_{\ell,r}$ is the probability that port ℓ is connected with server r . It holds that

$$\mathbb{P}\{\mathcal{D}_{\mathcal{K},I}\} = \prod_{\ell \in \mathcal{K}} \prod_{r \in I} (1 - z_{\ell,r}) \leq \exp \left(- \sum_{\ell \in \mathcal{K}} \sum_{r \in I} z_{\ell,r} \right) \leq \exp \left(- \sum_{\ell \in \mathcal{K}} \sum_{r \in I} \frac{\lambda_\ell H_j}{N\tilde{d}_j} \right), \quad (46)$$

and thus

$$\mathbb{P}\{\mathcal{D}_{\mathcal{K},I}\} \leq \exp \left(-|I| \frac{\sum_{\ell \in \mathcal{K}} \lambda_\ell H_j}{N\tilde{d}_j} \right) \leq \exp(-H_j Np_j) \leq 2^{-(N+L)}. \quad (47)$$

The first inequality is because $\ln(1+x) \leq x$ for $x > -1$, and $z_{\ell,r} < 1$. The second inequality is from the construction of G . The third inequality is from the definition of \mathcal{K} and \mathcal{I} . It thus holds that $\mathbb{P}\{C_j\} \leq 2^{N+L} 2^{-2(N+L)} = 2^{-(N+L)}$ by the union bound. Use the union bound once again, it holds $\mathbb{P}\{C_1 \cup C_2\} \leq 2^{-(N+L-1)}$.

For the total number of edges used in G_N , recall the definition of $p_1, p_2, \tilde{d}_1, \tilde{d}_2$ for a particular system in Assumption 2, and H_1, H_2 in Theorem 3.5. It holds that $\frac{\tilde{d}_1}{H_1} = O(\frac{\epsilon^2}{b^5(N+L)/N})$, and $\frac{\tilde{d}_2}{H_2} = O(\frac{\epsilon^2}{b^5(N+L)/N})$. Note that there are four types of connections on graph G_N as per Theorem 3.5, we bound their numbers of edges separately. First, the number of ports with $\lambda_\ell \geq N \frac{\tilde{d}_1}{H_1}$ is bounded by $\frac{N\mu_1 H_1}{Nd_1} = O(\frac{b^5(N+L)}{\epsilon^2} N)$ because $\lambda_\Sigma \leq N\mu_1$. Therefore, the number of connections from them is bounded by $O(\frac{b^5(N+L)}{\epsilon^2})$ since there are N servers. The same result holds for ports with $\lambda_\ell \geq N \frac{\tilde{d}_2}{H_2}$. Now for the remaining ports, the expected number of edges is upper bounded by $2 \sum_{\ell \in \mathcal{L}} \frac{\lambda_\ell}{N} \left(\frac{H_1}{d_1} + \frac{H_2}{d_2} \right) N = O\left(\frac{b^5(N+L)}{\epsilon^2}\right)$. Then to sum up, the expected number of edges in G_N scales as $O\left(\frac{b^5(N+L)}{\epsilon^2}\right)$. \square

6 SIMULATION RESULTS

In this section, we present simulation results for JFSQ and JFIQ. In particular, the following two settings are explored:

- we compare the mean response time of JFSQ, JFIQ with a recent paper [19] in a fixed-size system;
- we study the convergence of JFSQ and JFIQ on a random bipartite graph in the many-server regime.

We will also compare our policies with JSQ and JIQ where we assume that ties in those policies are broken at random. Detailed results are as follows.

6.1 Performance in a Fixed-Size System

We first study one particular setting as in [19]. There are 100 servers with fast service rate $\frac{25}{9}$, and 400 servers with slow service rate $\frac{5}{9}$. Jobs arrive into the system in a Poisson process of rate λ_Σ , and can be routed to any server. We simulate an infinite buffer system by setting the buffer size at each server to 10^6 . We compare JFSQ and JFIQ with JSQ, JIQ and JSQ-(2,2) introduced in [19]. JSQ-(2,2) is similar to Pod, and it is shown in [19] to perform better than other algorithms in light traffic. In addition, the communication overhead of JSQ-(2,2) is also as low as that of JFIQ. We refer the reader to the appendix for a detailed description of JSQ-(2,2). Besides, the lower bound result in Theorem 3.1 is plotted as a baseline. Define the system load to be $\frac{\lambda_\Sigma}{500}$. By increasing the system load, we can obtain Fig. 2. Clearly, Fig. 2 shows that JFSQ and JFIQ can achieve consistently fast mean response (very close to the lower bound) ranging from light traffic to heavy traffic (the system load is around 0.98). For other policies, JSQ-(2,2) performs well in light traffic. However, JIQ and JSQ could have relatively poor response time in light traffic, although JIQ is shown to have asymptotically zero waiting time [44].

6.2 Convergence in the Many-Server Regime

Next we explore the convergence behavior of JFSQ and JFIQ when there are job-server constraints. In particular, suppose there are N servers in the system. We assume there are four types of servers with the same amount of each type. Service times are all exponentially distributed, but with different

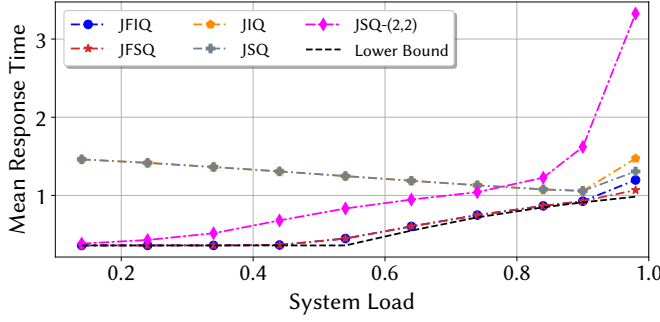


Fig. 2. The Mean Response Time of Different Routing Policies in a Fixed-Size System with Increasing System Load

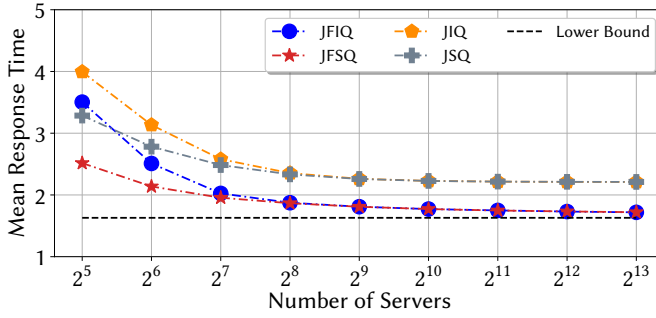


Fig. 3. The Mean Response Time of Different Routing Policies on Increasing-Sized Random Bipartite Graphs

service rate such that $\mu_i = 2^{-i+1}$, $i = 1, 2, 3, 4$. We also study the convergence of JSQ and JIQ. JSQ-(2,2) introduced above is not studied because it is designed for systems with two classes of servers.

The number of ports is set as $L = N^{1.5}$. The arrival rate to each port is assumed to be homogeneous, and is equal to $\frac{\lambda_\Sigma}{L}$ with $\lambda_\Sigma = 0.9 \sum_{i=1}^4 \frac{N\mu_i}{4}$. Denote the system load as $\lambda = 0.9$. In the corresponding bipartite graph, each port connects with each server with probability $\frac{2\sqrt{\ln N}}{N(1-\lambda)} \ln \frac{1}{1-\lambda}$ according to Theorem 3.6. The buffer size in this case is set as $b = 5$ because in many-server systems, we expect there to be little queueing and one should not need a large buffer size. Fig. 3 presents the convergence behavior of the mean-response time for JFSQ, JFIQ, JIQ and JSQ. It is interesting to notice that both JIQ and JFIQ suffer from slow mean response time when the system is small. But when the number of servers is $2^{11} = 2048$, the mean response time of JFSQ and JFIQ is very close to the lower bound. Such requirement on the number of servers is fine since modern cloud platforms can easily possess tens of thousands of servers [2]. On the other hand, both JSQ and JIQ also converge as N increases. Nevertheless, their mean response time is not optimal because they neglect server heterogeneity. Note that when the system is large, the blocking probability is nearly zero, even with a small buffer size. The convergence of the blocking probability is provided in the appendix. The setting is also extended to hyper-exponential service time distribution. For this new distribution, we show that although JFSQ and JFIQ have slow mean response times initially, their convergence behavior is similar to Fig. 3 when N increases. We refer the reader to the appendix for details.

7 CONCLUSION

In this paper, we studied the performance of two load balancing policies, JFSQ and JFIQ for load balancing on a bipartite graph. For a "well-connected" bipartite graph, we presented a bound on the mean response time for finite-size systems, which implies asymptotic optimality in the mean response time in both the many-server regime and the sub Halfin-Whitt regime. A by-product of this paper is a novel technique for bounding the distance to the mean-field limit of heterogeneous load balancing systems. In the analysis, we established three state-space collapse results to show that the system behaves similar to its mean-field limit. We also presented how to construct a sparse "well-connected" bipartite graph, where each left node is only connected to $\omega(\frac{1}{(1-\lambda)^2})$ right nodes when arrival rates are heterogeneous, and only $\omega(\frac{1}{1-\lambda} \ln \frac{1}{1-\lambda})$ nodes for homogeneous servers, given that the buffer size is a constant, and the number of left nodes is at least that of right nodes. However, it is unknown whether these two bounds are tight. In addition, a more general model than ours is to allow the service speed to depend on both the job types and the servers. Analyzing the performance of such a model is still an open problem, which we leave for future research.

Acknowledgment: The work of Wentao Weng was conducted during a visit to the Coordinated Science Lab, UIUC during 2020. Research is also supported in part by NSF Grants CPS ECCS 17-39189, CCF 19-34986, ECCS 16-09370, ONR Grant Navy N00014-19-1-2566, ARO Grant W911NF-19-1-0379.

REFERENCES

- [1] Amazon. 2020. *Amazon Web Services (AWS) Cloud Computing Services*. Retrieved August 3, 2020 from <https://aws.amazon.com>
- [2] George Amvrosiadis, Jun Woo Park, Gregory R Ganger, Garth A Gibson, Elisabeth Baseman, and Nathan DeBardeleben. 2018. On the diversity of cluster workloads and its impact on research results. In *Proc. USENIX Ann. Technical Conf. (ATC)*. 533–546.
- [3] Rami Atar. 2012. A diffusion regime with nondegenerate slowdown. *Operations Research* 60, 2 (2012), 490–500.
- [4] Sayan Banerjee and Debankur Mukherjee. 2019. Join-the-shortest queue diffusion limit in halfin-whitt regime: Tail asymptotics and scaling of extrema. *Ann. Appl. Probab.* 29, 2 (2019), 1262–1309.
- [5] Dimitris Bertsimas, David Gamarnik, and John N. Tsitsiklis. 2001. Performance of Multiclass Markovian Queueing Networks Via Piecewise Linear Lyapunov Functions. *Ann. Appl. Probab.* 11, 4 (11 2001), 1384–1428.
- [6] Anton Braverman. 2020. Steady-state analysis of the join-the-shortest-queue model in the halfin-whitt regime. *Math. Oper. Res.* (2020).
- [7] Anton Braverman, JG Dai, and Jiekun Feng. 2017. Stein's method for steady-state diffusion approximations: an introduction through the Erlang-A and Erlang-C models. *Stochastic Systems* 6, 2 (2017), 301–366.
- [8] Amarjit Budhiraja, Debankur Mukherjee, and Ruoyu Wu. 2019. Supermarket model on graphs. *The Annals of Applied Probability* 29, 3 (2019), 1740–1777.
- [9] Ellen Cardinaels, Sem Borst, and Johan S. H. van Leeuwen. 2020. Redundancy Scheduling with Locally Stable Compatibility Graphs. arXiv:2005.14566 [math.PR]
- [10] Ellen Cardinaels, Sem C Borst, and Johan SH van Leeuwen. 2019. Job assignment in large-scale service systems with affinity relations. *Queueing Syst.* 93, 3-4 (2019), 227–268.
- [11] James Cruise, Matthieu Jonckheere, and Seva Shneer. 2020. Stability of JSQ in queues with general server-job class compatibilities. *Queueing Syst.* (2020), 1–9.
- [12] Jeffrey Dean and Luiz André Barroso. 2013. The tail at scale. *Commun. ACM* 56, 2 (2013), 74–80.
- [13] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1 (2008), 107–113.
- [14] Atila Eryilmaz and R. Srikant. 2012. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Syst.* 72, 3-4 (2012), 311–359.
- [15] Patrick Eschenfeldt and David Gamarnik. 2018. Join the shortest queue with many servers. The heavy-traffic asymptotics. *Math. Oper. Res.* 43, 3 (2018), 867–886.
- [16] Serguei Foss and Natalia Chernova. 1998. On the stability of a partially accessible multi-station queue with state-dependent routing. *Queueing Syst.* 29, 1 (1998), 55–73.
- [17] David Gamarnik, John N Tsitsiklis, and Martin Zubeldia. 2018. Delay, memory, and messaging tradeoffs in distributed service systems. *Stoch. Syst.* 8, 1 (2018), 45–74.

- [18] David Gamarnik, John N Tsitsiklis, and Martin Zubeldia. 2020. A lower bound on the queueing delay in resource constrained load balancing. *Annals of Applied Probability* 30, 2 (2020), 870–901.
- [19] Kristen Gardner, Jazeem Abdul Jaleel, Alexander Wickeham, and Sherwin Doroudi. 2020. Scalable Load Balancing in the Presence of Heterogeneous Servers. *arXiv preprint arXiv:2006.13987* (2020).
- [20] Kristen Gardner and Rhonda Righter. 2020. Product Forms for FCFS Queueing Models with Arbitrary Server-Job Compatibilities: An Overview. *arXiv preprint arXiv:2006.05979* (2020).
- [21] Nicolas Gast. 2015. The Power of Two Choices on Graphs: the Pair-Approximation is Accurate? *ACM SIGMETRICS Performance Evaluation Review* 43, 2 (2015), 69–71.
- [22] Google. 2020. *Google Cloud Cloud Computing Services*. Retrieved August 3, 2020 from <https://cloud.google.com>
- [23] Google. 2020. *Google Search*. Retrieved August 3, 2020 from <https://www.google.com/search>
- [24] Arpan Gujarati, Sameh Elnikety, Yuxiong He, Kathryn S McKinley, and Björn B Brandenburg. 2017. Swayam: distributed autoscaling to meet SLAs of machine learning inference services with resource efficiency. In *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference*. 109–120.
- [25] Varun Gupta and Neil Walton. 2019. Load balancing in the nondegenerate slowdown regime. *Operations Research* 67, 1 (2019), 281–294.
- [26] Itai Gurvich. 2014. Diffusion models and steady-state approximations for exponentially ergodic Markovian queues. *The Annals of Applied Probability* 24, 6 (2014), 2527–2559.
- [27] Bruce Hajek. 1982. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability* (1982), 502–525.
- [28] Daniela Hurtado-Lange and Siva Theja Maguluri. 2020. Load balancing system under join the shortest queue: Many-server-heavy-traffic asymptotics. *arXiv preprint arXiv:2004.04826* (2020).
- [29] Daniela Hurtado-Lange and Siva Theja Maguluri. 2020. Throughput and Delay Optimality of Power-of-d Choices in Inhomogeneous Load Balancing Systems. *arXiv preprint arXiv:2004.00538* (2020).
- [30] Xin Liu, Kang Gong, and Lei Ying. 2020. Steady-State Analysis of Load Balancing with Coxian-2 Distributed Service Times. *arXiv preprint arXiv:2005.09815* (2020).
- [31] Xin Liu and Lei Ying. 2018. On achieving zero delay with power-of-d-choices load balancing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 297–305.
- [32] Xin Liu and Lei Ying. 2019. On universal scaling of distributed queues under load balancing. *arXiv preprint arXiv:1912.11904* (2019).
- [33] Xin Liu and Lei Ying. 2020. Steady-state analysis of load-balancing algorithms in the sub-Halfin-Whitt regime. *J. Appl. Probab.* 57, 2 (2020), 578–596.
- [34] 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.
- [35] Siva Theja Maguluri and R Srikant. 2016. Heavy traffic queue length behavior in a switch under the MaxWeight algorithm. *Stochastic Systems* 6, 1 (2016), 211–250.
- [36] Microsoft. 2020. *Microsoft Azure Cloud Computing Services*. Retrieved August 3, 2020 from <https://azure.microsoft.com/en-us/>
- [37] Michael Mitzenmacher. 2001. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems* 12, 10 (2001), 1094–1104.
- [38] Sharayu Moharir, Sujay Sanghavi, and Sanjay Shakkottai. 2015. Online load balancing under graph constraints. *IEEE/ACM Transactions on Networking* 24, 3 (2015), 1690–1703.
- [39] Debankur Mukherjee, Sem C Borst, and Johan SH Van Leeuwaarden. 2018. Asymptotically optimal load balancing topologies. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1 (2018), 1–29.
- [40] Debankur Mukherjee, Sem C Borst, Johan SH Van Leeuwaarden, and Philip A Whiting. 2018. Universality of power-of-d load balancing in many-server systems. *Stoch. Syst.* 8, 4 (2018), 265–292.
- [41] Kay Ousterhout, Patrick Wendell, Matei Zaharia, and Ion Stoica. 2013. Sparrow: distributed, low latency scheduling. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*. 69–84.
- [42] Daan Ruten and Debankur Mukherjee. 2020. Load Balancing Under Strict Compatibility Constraints. (2020).
- [43] Scott Shenker and Abel Weinrib. 1989. The optimal control of heterogeneous queueing systems: A paradigm for load-sharing and routing. *IEEE Trans. Comput.* 38, 12 (1989), 1724–1735.
- [44] Alexander L Stolyar. 2015. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Syst.* 80, 4 (2015), 341–361.
- [45] Alexander L Stolyar. 2015. Tightness of stationary distributions of a flexible-server system in the Halfin-Whitt asymptotic regime. *Stochastic Systems* 5, 2 (2015), 239–267.
- [46] John N. Tsitsiklis and Kuang Xu. 2017. Flexible Queueing Architectures. *Oper. Res.* 65, 5 (2017), 1398–1413. <https://doi.org/10.1287/opre.2017.1620> arXiv:<https://doi.org/10.1287/opre.2017.1620>

- [47] Stephen RE Turner. 1998. The effect of increasing routing choice on resource pooling. *Probability in the Engineering and Informational Sciences* 12, 1 (1998), 109–124.
- [48] Jeremy Visschers, Ivo Adan, and Gideon Weiss. 2012. A product form solution to a system with multi-type jobs and multi-type servers. *Queueing Syst.* 70, 3 (2012), 269–298.
- [49] 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.
- [50] Weina Wang, Siva Theja Maguluri, R Srikant, and Lei Ying. 2018. Heavy-traffic delay insensitivity in connection-level models of data transfer with proportionally fair bandwidth sharing. In *Proc. ACM SIGMETRICS Int. Conf. Measurement and Modeling of Computer Systems*, Vol. 45. ACM, 232–245.
- [51] Weina Wang, Kai Zhu, Lei Ying, Jian Tan, and Li Zhang. 2014. Maptask scheduling in mapreduce with data locality: Throughput and heavy-traffic optimality. *IEEE/ACM Transactions On Networking* 24, 1 (2014), 190–203.
- [52] Richard R Weber. 1978. On the optimal assignment of customers to parallel servers. *J. Appl. Probab.* 15, 2 (1978), 406–413.
- [53] Wentao Weng and Weina Wang. 2020. Dispatching Parallel Jobs to Achieve Zero Queuing Delay. *arXiv preprint arXiv:2004.02081* (2020).
- [54] Qiaomin Xie and Yi Lu. 2015. Priority algorithm for near-data scheduling: Throughput and heavy-traffic optimality. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 963–972.
- [55] Qiaomin Xie, Ali Yekkehkhany, and Yi Lu. 2016. Scheduling with multi-level data locality: Throughput and heavy-traffic optimality. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.
- [56] Lei Ying. 2017. Stein's method for mean field approximations in light and heavy traffic regimes. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 1 (2017), 1–27.
- [57] Lei Ying, R. Srikant, and Xiaohan Kang. 2017. The power of slightly more than one sample in randomized load balancing. *Math. Oper. Res.* 42, 3 (2017), 692–722.
- [58] Xingyu Zhou and Ness Shroff. 2020. A Note on Load Balancing in Many-Server Heavy-Traffic Regime. *arXiv preprint arXiv:2004.09574* (2020).
- [59] Xingyu Zhou, Jian Tan, and Ness Shroff. 2018. Flexible load balancing with multi-dimensional state-space collapse: Throughput and heavy-traffic delay optimality. *Performance Evaluation* 127 (2018), 176–193.
- [60] Xingyu Zhou, Jian Tan, and Ness Shroff. 2018. Heavy-traffic delay optimality in pull-based load balancing systems: Necessary and sufficient conditions. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 3 (2018), 1–33.

A IMPLEMENTATION OF JFSQ AND JFIQ

In the following, we discuss how to implement JFSQ and JFIQ and their associated time complexity and communication overhead.

We first consider JFSQ. Upon the arrival of a job at port ℓ , the dispatcher requests the queue lengths from all servers that are connected to port ℓ . Then we find the server with the shortest queue length. If there is a tie, we choose the one with the fastest service speed. In this way, the time complexity and the communication overhead per arrival is $O(|N_L(\ell)|)$, where $N_L(\ell)$ is the set of servers connected to port ℓ .

Now for JFIQ, we maintain N variables I_r , $1 \leq r \leq N$ in the memory of the dispatcher where $I_r = 1$ if server r is idle and $I_r = 0$ if server r is busy. Then when a job arrives at port ℓ , we find $r \in N_L(\ell)$ such that $I_r = 1$ and μ_r is the maximum. If there is no such r , we let r to be a random server in $N_L(\ell)$. We then route the job to server r . In addition, I_r is set as zero. Moreover, when a server becomes idle, it sends back a message to the dispatcher, and I_r is set as one. We next analyze the complexity of JFIQ. First, the time complexity per job is the same as JFSQ, and is equal to $O(|N_L(\ell)|)$. Second, we can see that the amortized number of messages per arrival between servers and the dispatcher is at most one (and is fact smaller than that). This is because an arbitrary arrival can incur at most one message. In a nutshell, for JFIQ, the time complexity per job is $O(|N_L(\ell)|)$ and the averaged number of messages per job is at most one.

B NOTATION TABLE

In Table 1, we summarize important notations used in the paper.

Table 1. Notation Table

Notation	Definition
\mathcal{L}	The set of ports
\mathcal{R}	The set of servers
E	Edges between ports and servers
L	The number of ports
N	The number of servers
$N_L(\ell)$	Set of servers connected with port ℓ
$N_R(r)$	Set of ports connected with server r
$N_R(\mathcal{I})$	Set of ports connected with at least one server in the set of servers \mathcal{I}
λ_ℓ	Arrival rate of jobs to port ℓ
λ_Σ	Total arrival rate of jobs to the system, given by $\lambda_\Sigma = \sum_{\ell=1}^L \lambda_\ell$
λ	Normalized total arrival rate, i.e., $\lambda = \frac{\lambda_\Sigma}{N}$
$D_{\mathcal{I}}$	Sum of arrival rates at ports not connected with at least one server in \mathcal{I}
b	Buffer size at each server
M	The number of types of servers
μ_m	Service rate of a server of type m
α_m	Fraction of type m servers
N_m	The number of type m servers, given by $N_m = N\alpha_m$
\mathbf{q}	State of queue lengths with q_r being the queue length of server r
$s_{m,i}(\mathbf{q})$	The number of type m servers with queue length at least i divided by N
$C_m(\mathbf{q})$	The number of jobs in type m servers divided by N
β	"Idle" fraction of service rate, i.e., $\lambda_\Sigma = N \sum_{m=1}^K \mu_m \alpha_m (1 - \beta)$
C_m^*	The number of busy type m servers in the mean-field limit divided by N
ϵ	Approximation error in analysis
τ_{ab}	Ratio of the service rate of a type a server over that of a type b server
p_1, p_2	Fraction of servers needed to test for well connectedness
\tilde{d}_1, \tilde{d}_2	Upper bound of disconnected arrival rates (divided by N)
$z_{\ell,r}$	Connecting probability of a port ℓ and a server r in a random graph construction

C PROOF OF PROPOSITION 3.1

PROPOSITION 3.1[RESTATED]. *Suppose that the buffer size is infinite, i.e. $b = \infty$. Let \bar{Z} be the random variable denoting the service time of an arbitrary job. Then for any stable policy, the mean number of jobs in the system is lower bounded by NC^* , and*

$$\mathbb{E}[\bar{Z}] \geq \frac{C^*}{\lambda}. \quad (48)$$

PROOF. For any $m \in \{1, \dots, M\}$, let I_m denote the probability that an arriving job is scheduled to a type- m server in steady state. Also, recall that $\bar{s}_{m,1}$ is defined as a steady-state random variable denoting the number of busy type- m servers divided by N . Then because of stability and work conservation law, it holds that for all $m \leq M$,

$$\lambda_\Sigma I_m = N \mu_m \mathbb{E}[\bar{s}_{m,1}]. \quad (49)$$

In particular,

$$\lambda = \sum_{m=1}^M \frac{\lambda_{\Sigma} I_m}{N} = \sum_{m=1}^M \mu_m \mathbb{E} [\bar{S}_{m,1}] \quad (50)$$

since $\sum_{m=1}^M I_m = 1$. Now notice that the mean service time of jobs is given by

$$\mathbb{E} [\bar{Z}] = \sum_{m=1}^M \frac{I_m}{\mu_m} = \sum_{m=1}^M \frac{\mathbb{E} [\bar{S}_{m,1}]}{\lambda} \quad (51)$$

since the service time at type- m servers is exponentially distributed with mean $\frac{1}{\mu_m}$, and I_m satisfies (49). To obtain a lower bound of $\mathbb{E} [\bar{Z}]$, consider the following linear programming problem.

$$\begin{aligned} \min \quad & \frac{1}{\lambda} \sum_{m=1}^M x_m \\ \text{s.t.} \quad & \lambda = \sum_{m=1}^M \mu_m x_m, \quad m = 1, \dots, M \\ & 0 \leq x_m \leq \alpha_m, \quad m = 1, \dots, M \end{aligned}$$

where x_m is an analog of $\mathbb{E} [\bar{S}_{m,1}]$, and the objective value is a lower bound of $\mathbb{E} [\bar{Z}]$ because of (50). Then since only the sum of x_m matters, and $\mu_1 \geq \dots \geq \mu_M$, the optimal solution is exactly given by $x_1^* = \alpha_1, \dots, x_{K-1}^* = \alpha_{K-1}, x_K^* = \frac{\lambda - \sum_{m=1}^{K-1} \mu_m x_m}{\mu_K}, x_m^* = 0$ for $m > K$. Then it is clear that $\mathbb{E} [\bar{Z}] \geq \frac{1}{\lambda} \sum_{m=1}^M x_m^* = \frac{C^*}{\lambda}$. \square

D PROOF OF LEMMAS IN SECTION 4

D.1 Proof of Lemma 4.2

LEMMA 4.2[RESTATE]. *The expectation $\mathbb{E} [Gg(\sum_{m=1}^K C_m(\bar{Q}))]$ is equal to 0.*

PROOF. To simplify the notation, denote $V(\mathbf{q}) = g(\sum_{m=1}^K C_m(\mathbf{q}))$ for a state \mathbf{q} . Now that since the system is stable (because of the assumption of finite buffers), there is a unique stationary distribution $\pi_{\mathbf{q}}$ that solves the balancing equation such that for every \mathbf{q} ,

$$\pi_{\mathbf{q}} \sum_{\mathbf{q}'} r_{\mathbf{q},\mathbf{q}'} = \sum_{\mathbf{q}'} \pi_{\mathbf{q}'} r_{\mathbf{q}',\mathbf{q}} \quad (52)$$

where $r_{\mathbf{q},\mathbf{q}'}$ is the transition rate from \mathbf{q} to \mathbf{q}' . Now that $V(\mathbf{q})$ is bounded (as $\sum_{m=1}^K C_m(\mathbf{q}) \leq b$), it holds

$$\begin{aligned} \mathbb{E} [GV(\bar{Q})] &= \sum_{\mathbf{q}} \pi_{\mathbf{q}} \sum_{\mathbf{q}'} r_{\mathbf{q},\mathbf{q}'} (V(\mathbf{q}') - V(\mathbf{q})) \\ &= - \sum_{\mathbf{q}} \pi_{\mathbf{q}} \sum_{\mathbf{q}'} V(\mathbf{q}) r_{\mathbf{q},\mathbf{q}'} + \sum_{\mathbf{q}} \pi_{\mathbf{q}} \sum_{\mathbf{q}'} r_{\mathbf{q},\mathbf{q}'} V(\mathbf{q}') \\ &= - \sum_{\mathbf{q}} V(\mathbf{q}) \sum_{\mathbf{q}'} \pi_{\mathbf{q}} r_{\mathbf{q},\mathbf{q}'} + \sum_{\mathbf{q}} V(\mathbf{q}) \sum_{\mathbf{q}'} \pi_{\mathbf{q}'} r_{\mathbf{q}',\mathbf{q}} \\ &= 0. \end{aligned}$$

\square

D.2 Proof of Lemma 4.3

LEMMA 4.3[RESTATED]. *It holds that*

$$\mathbb{E} \left[h \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) \right] \leq \mathbb{E} \left[\mathbb{1} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \geq \eta + \frac{1}{N} \right\} g' \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) (\lambda + \mu_1 \delta - W(\bar{\mathbf{Q}})) \right] + \frac{38b^2\tau_{1K}}{\epsilon N}. \quad (20)$$

PROOF. The idea is to utilize the result that $\mathbb{E} \left[h \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) \right] \leq (17)(18) + (19)$, and to expand (18) and (19) by Taylor's expansion. Consider three cases of state \mathbf{q} .

- First, if $\sum_{m=1}^K C_m(\mathbf{q}) \leq \eta - \frac{1}{N}$, then $g(\sum_{m=1}^K C_m(\mathbf{q}) - \frac{1}{N})$, $g(\sum_{m=1}^K C_m(\mathbf{q}))$, $g(\sum_{m=1}^K C_m(\mathbf{q}) + \frac{1}{N})$ are all zero. This case has no contribution to the expectation;
- second, if $\sum_{m=1}^K C_m(\mathbf{q}) \in (\eta - \frac{1}{N}, \eta + \frac{1}{N})$, by first-order Taylor's expansion, there exists some $\tilde{\xi}_{\mathbf{q}}, \tilde{\eta}_{\mathbf{q}} \in (\eta - \frac{2}{N}, \eta + \frac{2}{N})$, such that

$$\begin{aligned} g \left(\sum_{m=1}^K C_m(\mathbf{q}) + \frac{1}{N} \right) - g \left(\sum_{m=1}^K C_m(\mathbf{q}) \right) &= \frac{1}{N} g'(\tilde{\xi}_{\mathbf{q}}), \\ g \left(\sum_{m=1}^K C_m(\mathbf{q}) - \frac{1}{N} \right) - g \left(\sum_{m=1}^K C_m(\mathbf{q}) \right) &= -\frac{1}{N} g'(\tilde{\eta}_{\mathbf{q}}); \end{aligned}$$

- third, if $\sum_{m=1}^K C_m(\mathbf{q}) \geq \eta + \frac{1}{N}$, by second-order Taylor's expansion, there exists some $\xi_{\mathbf{q}}, \eta_{\mathbf{q}}$, such that

$$\begin{aligned} g \left(\sum_{m=1}^K C_m(\mathbf{q}) + \frac{1}{N} \right) - g \left(\sum_{m=1}^K C_m(\mathbf{q}) \right) &= \frac{1}{N} g' \left(\sum_{m=1}^K C_m(\mathbf{q}) \right) + \frac{2}{N^2} g''(\xi_{\mathbf{q}}), \\ g \left(\sum_{m=1}^K C_m(\mathbf{q}) - \frac{1}{N} \right) - g \left(\sum_{m=1}^K C_m(\mathbf{q}) \right) &= -\frac{1}{N} g' \left(\sum_{m=1}^K C_m(\mathbf{q}) \right) + \frac{2}{N^2} g''(\eta_{\mathbf{q}}). \end{aligned}$$

Then it holds that

$$\mathbb{E} \left[h \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) \right] \quad (53)$$

$$\leq (17) + (18) + (19) \quad (54)$$

$$= \mathbb{E} \left[\mathbb{1} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \geq \eta + \frac{1}{N} \right\} \left(g' \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) (\lambda + \mu_1 \delta - W(\bar{\mathbf{Q}})) \right) \right] \quad (55)$$

$$+ \mathbb{E} \left[\mathbb{1} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \geq \eta + \frac{1}{N} \right\} \left(\frac{2}{N} \left(\lambda g''(\xi_{\bar{\mathbf{Q}}}) + W(\bar{\mathbf{Q}}) g''(\eta_{\bar{\mathbf{Q}}}) \right) \right) \right] \quad (56)$$

$$+ \mathbb{E} \left[\mathbb{1} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \in (\eta - \frac{1}{N}, \eta + \frac{1}{N}) \right\} \left(g' \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \right) (\mu_1 \delta) + \lambda g'(\tilde{\xi}_{\bar{\mathbf{Q}}}) - W(\bar{\mathbf{Q}}) g'(\tilde{\eta}_{\bar{\mathbf{Q}}}) \right) \right]. \quad (57)$$

It suffices to bound (56) and (57). First, note that $|g''(x)| \leq \frac{1}{\mu_1 \delta}$ for all x by the explicit form of $g(x)$ in (11). It holds

$$(56) \leq \frac{2}{N} \cdot \frac{1}{\mu_1 \delta} \cdot 2\mu_1 = \frac{4}{N\delta} = \frac{24\tau_{1K}b^2}{\epsilon N}. \quad (58)$$

On the other hand, to bound (57), since $\sum_{m=1}^K C_m(\tilde{\mathbf{Q}})$, $\tilde{\xi}_{\mathbf{q}}$, $\tilde{\eta}_{\mathbf{q}} \in (\eta - \frac{2}{N}, \eta + \frac{2}{N})$, their derivatives are all bounded by $\frac{2}{N\mu_1\delta}$. Then

$$(57) \leq \frac{2}{N\mu_1\delta} \cdot (\mu_1\delta + \mu_1) = \frac{2}{N} + \frac{12\tau_{1K}b^2}{\epsilon N} \leq \frac{14\tau_{1K}b^2}{\epsilon N}. \quad (59)$$

Summing the above two equations completes the proof of Lemma 4.3. \square

D.3 Proof of Lemma 4.4

LEMMA 4.4[RESTATE]. Consider the following Lyapunov function

$$V_1(\mathbf{q}) = \min \left(\sum_{j=1}^b s_{K,j}(\mathbf{q}) + \sum_{m=1}^{K-1} \sum_{j=2}^b s_{m,j}(\mathbf{q}), \sum_{m=1}^{K-1} C_m^* - \sum_{m=1}^{K-1} s_{m,1}(\mathbf{q}) \right). \quad (22)$$

It holds that if $V_1(\mathbf{q}) \geq B_1 := \tau_{1K}\delta$, then $GV_1(\mathbf{q}) \leq \frac{-\mu_1\delta}{2b}$.

PROOF. Since $V_1(\mathbf{q}) \geq B_1$ by assumption, both of the following two properties holds:

$$\sum_{j=1}^b s_{K,j}(\mathbf{q}) + \sum_{m=1}^{K-1} \sum_{j=2}^b s_{m,j}(\mathbf{q}) \geq B_1; \quad (60)$$

$$\sum_{m=1}^{K-1} s_{m,1}(\mathbf{q}) \leq \sum_{m=1}^{K-1} C_m^* - B_1. \quad (61)$$

Let $\mathcal{T}_{1,1}$ be the first term in $V_1(\mathbf{q})$, and $\mathcal{T}_{1,2}$ be the second term. First, by definition,

$$\begin{aligned} GV_1(\mathbf{q}) &= \sum_{\mathbf{q}'} r_{\mathbf{q},\mathbf{q}'} (V_1(\mathbf{q}') - V_1(\mathbf{q})) \\ &= \sum_{\mathbf{q}', \text{arrival}} r_{\mathbf{q},\mathbf{q}'} (V_1(\mathbf{q}') - V_1(\mathbf{q})) \end{aligned} \quad (62)$$

$$+ \sum_{\mathbf{q}', \text{departure}} r_{\mathbf{q},\mathbf{q}'} (V_1(\mathbf{q}') - V_1(\mathbf{q})) \quad (63)$$

where we separate transitions by identifying those caused by a job arrival from those caused by a job departure. Bounding (62) and (63) can then bound $GV_1(\mathbf{q})$. Next we consider two cases corresponding to whether $V_1(\mathbf{q})$ is equal to $\mathcal{T}_{1,1}$ or to $\mathcal{T}_{1,2}$.

Suppose that $\mathcal{T}_{1,1} \leq \mathcal{T}_{1,2}$, then in this case,

$$(63) \leq - \left(\sum_{j=1}^b \mu_K (s_{K,j}(\mathbf{q}) - s_{K,j+1}(\mathbf{q})) + \sum_{m=1}^{K-1} \sum_{j=2}^b \mu_m (s_{m,j}(\mathbf{q}) - s_{m,j+1}(\mathbf{q})) \right) \quad (64)$$

$$= - \left(\mu_K s_{K,1}(\mathbf{q}) + \sum_{m=1}^K \mu_m s_{m,2}(\mathbf{q}) \right) \quad (65)$$

$$\leq - \frac{B_1 \mu_K}{b} \leq \frac{-\mu_1\delta}{b}. \quad (66)$$

The first inequality (64) is because $V_1(\mathbf{q}) = \mathcal{T}_{1,1}$, and only jobs departing from servers of type K and servers of types less than K with queue length at least 2 can affect the value of $V_1(\mathbf{q})$. The first equation (65) comes from the fact that $s_{m,b+1} = 0$ for all m . The last inequality is from (60) and the non-decreasing property

$$s_{m,1}(\mathbf{q}) \geq s_{m,2}(\mathbf{q}) \geq \dots s_{m,b}(\mathbf{q})$$

for all m .

On the other hand, to bound (62), notice that $V_1(\mathbf{q})$ can increase only when a job arrival is routed to some servers of types at least K . Then clearly,

$$(62) \leq \sum_{\ell=1}^L \frac{1}{N} \lambda_{\ell} \cdot \mathbb{1} \{ \text{an arrival to port } \ell \text{ is routed to an idle server of types at least } k \mid \mathbf{q} \}. \quad (67)$$

However, by (61), the number of idle servers of types less than K is at least

$$N \sum_{m=1}^{K-1} (C_m^* - s_{m,1}(\mathbf{q})) \geq NB_1 = \frac{N\epsilon}{6b^2}.$$

Let \mathcal{I} be the set of idle servers of types less than K . Since $|\mathcal{I}| \geq \frac{N\epsilon}{6b^2}$, Assumption 2 guarantees that $\sum_{\ell \notin N_{\mathcal{R}}(\mathcal{I})} \lambda_{\ell} \leq N\tilde{d}_1 = \frac{N\epsilon\mu_K}{12b^3}$. That is to say, the total arrival rates of ports not connected with servers in \mathcal{I} is bounded by $N\tilde{d}_1$. Now since our routing policy is either JFSQ or JFIQ, for those ports connected with \mathcal{I} , a job arrival must be routed to one server in \mathcal{I} because servers in \mathcal{I} are idle, and are faster than other idle servers not in \mathcal{I} . Therefore,

$$(67) \leq \frac{1}{N} \cdot \frac{N\epsilon\mu_K}{12b^3} \leq \frac{\mu_1\delta}{2b}. \quad (68)$$

With (66) and (68), it holds $GV_1(\mathbf{q}) \leq \frac{-\mu_1\delta}{2b}$ when $\mathcal{T}_{1,1} \leq \mathcal{T}_{1,2}$.

For the second case where $\mathcal{T}_{1,1} \geq \mathcal{T}_{1,2}$, it holds

$$(63) \leq \sum_{m=1}^{K-1} \mu_m (s_{m,1}(\mathbf{q}) - s_{m,2}(\mathbf{q})) \quad (69)$$

since $V_1(\mathbf{q})$ increases only when a job departs from a server of type less than K and only with this single job in the server. Also, we can see

$$(62) \leq -\frac{1}{N} \sum_{\ell=1}^L \lambda_{\ell} \cdot \mathbb{1} \{ \text{an arrival to port } \ell \text{ is routed to an idle server of type less than } k \mid \mathbf{q} \} \quad (70)$$

$$\leq \frac{1}{N} (-\lambda_{\Sigma} + N\tilde{d}_1) = -\lambda + \tilde{d}_1. \quad (71)$$

The first inequality is because for arrival transitions, only jobs arriving to idle servers of types less than k can change $V_1(\mathbf{q})$, and their arrivals will all decrease $V_1(\mathbf{q})$ by $\frac{1}{N}$ by the definition of $\mathcal{T}_{1,2}$. The second inequality is derived from the same argument of (68). Therefore, it holds that

$$GV_1(\mathbf{q}) = (62) + (63) \leq -\lambda + \tilde{d}_1 + \sum_{m=1}^{K-1} \mu_m (s_{m,1}(\mathbf{q}) - s_{m,2}(\mathbf{q})) \leq -\lambda + \tilde{d}_1 + \sum_{m=1}^{K-1} \mu_m \alpha_m - \mu_K B_1 \quad (72)$$

$$\leq -\mu_K B_1 + \tilde{d}_1 \quad (73)$$

$$\leq -\frac{\mu_1\delta}{2b} \quad (74)$$

because of (61) and the assumption that $\lambda \geq \sum_{m=1}^{K-1} \mu_m \alpha_m$.

Therefore, the above discussion proves that whenever $V_1(\mathbf{q}) \geq B_1$, it holds $GV_1(\mathbf{q}) \leq -\frac{\mu_1\delta}{2b}$. \square

D.4 Proof of Lemma 4.5

LEMMA 4.5[RESTATED]. Consider the following Lyapunov function

$$V_2(\mathbf{q}) = \min \left(\sum_{m=1}^K \sum_{j=2}^b s_{m,j}(\mathbf{q}), \sum_{m=1}^K C_m^* + B_2 + 3\tau_{1K}\bar{\delta} - \sum_{m=1}^K s_{m,1}(\mathbf{q}) \right) \quad (23)$$

where $\bar{\delta} := \tau_{1K}\delta$, and $B_2 := \frac{1}{2}\epsilon + \bar{\delta}$. It holds that if $V_2(\mathbf{q}) \geq B_2$, then $GV_2(\mathbf{q}) \leq -\frac{\mu_1\delta}{b}$.

PROOF. Let $\mathcal{T}_{2,1}$ be the first term in $V_2(\mathbf{q})$, and $\mathcal{T}_{2,2}$ be the second term. Since $V_2(\mathbf{q}) \geq B_2$, both the following hold:

$$\sum_{m=1}^K \sum_{j=2}^b s_{ij}(\mathbf{q}) \geq B_2; \quad (75)$$

$$\sum_{m=1}^K s_{m,1}(\mathbf{q}) \leq \sum_{m=1}^K C_m^i + 3\mu\bar{\delta}. \quad (76)$$

By definition,

$$GV_2(\mathbf{q}) = \sum_{\mathbf{q}', \text{arrival}} r_{\mathbf{q}, \mathbf{q}'} (V_2(\mathbf{q}') - V_2(\mathbf{q})) \quad (77)$$

$$+ \sum_{\mathbf{q}', \text{departure}} r_{\mathbf{q}, \mathbf{q}'} (V_2(\mathbf{q}') - V_2(\mathbf{q})). \quad (78)$$

We then consider two cases. First, suppose that $\mathcal{T}_{2,1} \leq \mathcal{T}_{2,2}$. Then similar to the proof of Lemma 4.4, using (75), it holds that

$$(78) \leq -\frac{1}{N} \sum_{m=1}^K \sum_{j=2}^b N\mu_m (s_{m,j}(\mathbf{q}) - s_{m,j+1}(\mathbf{q})) \quad (79)$$

$$= -\frac{1}{N} \sum_{m=1}^K N\mu_m s_{m,2}(\mathbf{q}) \quad (80)$$

$$\leq -\frac{B_2\mu_K}{b} = -\frac{\epsilon\mu_K}{2b} - \frac{\mu_1\delta}{b}. \quad (81)$$

On the other hand, we have

$$(77) \leq \sum_{\ell=1}^L \frac{1}{N} \lambda_\ell \cdot \mathbb{1} \{ \text{an arrival to port } \ell \text{ is routed to an idle server of types } \geq k+1 \mid \mathbf{q} \}. \quad (82)$$

Notice that by (76), the number of idle servers of types no greater than K satisfies that

$$N \left(\sum_{m=1}^K \alpha_m - \sum_{m=1}^K s_{m,1}(\mathbf{q}) \right) \quad (83)$$

$$\geq N \left(\sum_{m=1}^K \alpha_m - \sum_{m=1}^K C_m^* - 3\tau_{1,K}\bar{\delta} \right) \quad (84)$$

$$= N \left(\alpha_K - \frac{\lambda - \sum_{m=1}^{K-1} \mu_m \alpha_m}{\mu_K} - 3\tau_{1,K}\bar{\delta} \right) \quad (85)$$

$$= N \cdot \frac{\sum_{m=1}^K \mu_m \alpha_m - \lambda}{\mu_K} - 3N\tau_{1,K}\bar{\delta} \quad (86)$$

$$= \frac{N}{\mu_K} \left(\beta \sum_{m=1}^K \mu_m \alpha_m - 3\mu_1 \tau_{1K} \delta \right) \quad (87)$$

$$\geq N \left(\hat{\beta} - 3\tau_{1K} \frac{\epsilon}{6b^2} \right) \geq \frac{N\hat{\beta}}{2} \quad (88)$$

where (88) is because $b^2 \geq \tau_{1K}$ by Assumption 1, and $\hat{\beta} = \beta \sum_{m=1}^K \alpha_m$, and $\mu_1 > \dots > \mu_K$.

Let \mathcal{I} be the set of idle servers of types no greater than K . It then holds $|\mathcal{I}| \geq \frac{N\hat{\beta}}{2}$. Then By Assumption 2, the total arrival rate of ports not connected with \mathcal{I} is bounded by $N\tilde{d}_2$. Since the routing policy is either JFSQ or JIFQ, jobs arriving to ports connecting with \mathcal{I} must be routed to servers in \mathcal{I} . Therefore, it holds (82) $\leq \tilde{d}_2 \leq \frac{\mu_K \epsilon}{2b}$. Then in this case, we know

$$GV_2(\mathbf{q}) = (77) + (78) \leq -\frac{\epsilon\mu_K}{2b} - \frac{\mu_1\delta}{b} + \frac{\mu_K\epsilon}{2b} \leq -\frac{\mu_1\delta}{b}.$$

Now we consider the second case, $\mathcal{T}_{2,1} \geq \mathcal{T}_{2,2}$. Similarly, it holds (78) $\leq \sum_{m=1}^K \mu_m (s_{m,1}(\mathbf{q}) - s_{m,2}(\mathbf{q}))$, and

$$\begin{aligned} (77) &\leq -\frac{1}{N} \sum_{\ell=1}^L \frac{1}{N} \lambda_\ell \cdot \mathbb{1} \{ \text{an arrival to port } \ell \text{ is routed to an idle server of types } \leq k \mid \mathbf{q} \} \\ &\leq -\lambda + \tilde{d}_2 \end{aligned} \quad (89)$$

where the last inequality follows the same argument as in the first case. Then it holds

$$GV_2(\mathbf{q}) \leq \sum_{m=1}^K \mu_m s_{m,1}(\mathbf{q}) - \sum_{m=1}^K \mu_m s_{m,2}(\mathbf{q}) - \lambda + \tilde{d}_2 \quad (90)$$

$$\leq \sum_{m=1}^{K-1} \mu_m \alpha_m + \mu_K (C_K^* + 3\mu_1 \delta) - \lambda - \frac{\mu_K B_2}{b-1} + \frac{\mu_K \epsilon}{2b} \quad (91)$$

$$\leq 3\mu_1 \delta - \frac{\mu_K B_2}{b-1} + \frac{\epsilon}{2b} \quad (92)$$

$$\leq 3\mu_1 \delta - \frac{\mu_K \epsilon}{2(b-1)} + \frac{\mu_K \epsilon}{2b} - \frac{\mu_1 \delta}{b} \quad (93)$$

$$\leq -\frac{\mu_1 \delta}{b}. \quad (94)$$

The last inequality is because

$$\frac{\mu_K \epsilon}{2(b-1)} - \frac{\mu_K \epsilon}{2b} = \frac{\mu_K \epsilon}{2b^2} \geq 3\mu_1 \frac{\mu_K \epsilon}{6\mu_1 b^2} = 3\mu_1 \delta.$$

Therefore, we complete the proof of Lemma 4.5. \square

D.5 Proof of Lemma 4.10

LEMMA 4.10[RESTATED]. For any $\Delta \geq \frac{\hat{\beta}}{2}$, it holds $\mathbb{P}\{\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) > C^* + \Delta\} \leq \frac{104\tau_{1K}b^2}{\Delta\epsilon N}$.

PROOF. By Lemma 4.1, it holds that

$$\mathbb{P}\left\{\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) > C^* + \Delta\right\} = \mathbb{P}\left\{\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) - C^* - \frac{\hat{\beta}}{4} > \Delta - \frac{\hat{\beta}}{4}\right\} \quad (95)$$

$$\leq \mathbb{P}\left\{\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) - C^* - \frac{\hat{\beta}}{4} > \frac{1}{2}\Delta\right\} \quad (96)$$

$$\leq \frac{\mathbb{E} \left[\max \left(\sum_{m=1}^K C_m(\bar{\mathbf{Q}}) - C^* - \frac{\hat{\beta}}{4}, 0 \right) \right]}{\frac{1}{2}\Delta} \quad (97)$$

$$\leq \frac{208\tau_{1K}b^2}{\Delta\epsilon N} \quad (98)$$

since $\epsilon \leq \frac{\hat{\beta}}{4}$ by assumption. \square

D.6 Proof of Lemma 4.11

LEMMA 4.11[RESTATE]. When $V_3(\mathbf{q}) \geq B_3$, it holds that

- if $\mathbf{q} \in \mathcal{E}_K$, the drift is bounded as $GV_3(\mathbf{q}) \leq -\frac{B_3\mu_M}{b} + \tilde{d}_2$;
- if $\mathbf{q} \notin \mathcal{E}_K$, the drift is bounded as $GV_3(\mathbf{q}) \leq \mu_1$.

PROOF. By definition,

$$\begin{aligned} GV_3(\mathbf{q}) &= \sum_{\mathbf{q}'} r_{\mathbf{q},\mathbf{q}'} (V_3(\mathbf{q}') - V_3(\mathbf{q})) \\ &= \sum_{\mathbf{q}', \text{arrival}} r_{\mathbf{q},\mathbf{q}'} (V_3(\mathbf{q}') - V_3(\mathbf{q})) \end{aligned} \quad (99)$$

$$+ \sum_{\mathbf{q}', \text{departure}} r_{\mathbf{q},\mathbf{q}'} (V_3(\mathbf{q}') - V_3(\mathbf{q})). \quad (100)$$

Note that since $V_3(\mathbf{q}) \geq B_3$, and $V_3(\mathbf{q}) = \sum_{m=K+1}^M \sum_{j=1}^b s_{m,j}(\mathbf{q})$, it holds that

$$(100) = - \sum_{m=K+1}^M \mu_m s_{m,1}(\mathbf{q}) \geq -\frac{B_3\mu_M}{b} \quad (101)$$

since $s_{m,1}(\mathbf{q}) \geq \dots \geq s_{m,b}(\mathbf{q})$ and $s_{m,b+1}(\mathbf{q}) = 0$ for all m .

For (99), we consider two cases. First, if $\mathbf{q} \in \mathcal{E}_K$, the number of idle servers of types no greater than K is given by

$$\begin{aligned} &N \left(\sum_{m=1}^K \alpha_m - \sum_{m=1}^K s_{m,1}(\mathbf{q}) \right) \\ &\geq N \left(\sum_{m=1}^K \alpha_m - \sum_{m=1}^K C_m(\mathbf{q}) \right) \\ &\geq N \left(\sum_{m=1}^K \alpha_m - C^* - \frac{\hat{\beta}}{2} \right) \\ &= N \left(\frac{\beta \sum_{m=1}^{K-1} \alpha_m \mu_m}{\mu_K} - \frac{\hat{\beta}}{2} \right) \\ &\geq N \frac{\hat{\beta}}{2} \end{aligned}$$

where the second inequality is because $\sum_{m=1}^K C_m(\mathbf{q}) \leq C^* + \frac{\hat{\beta}}{2}$ when $\mathbf{q} \in \mathcal{E}_K$. Then since the routing policy is either JFSQ or JFIQ, jobs arriving to ports connecting with idle servers of types no greater than K must be routed to those servers. And by Assumption 2, the total arrival rate of disconnected ports is bounded by $\tilde{d}_2 N$. As a result,

$$(99) \leq \tilde{d}_2, \quad (102)$$

showing that $GV_3(\mathbf{q}) \leq -\frac{B_3\mu_M}{b} + \tilde{d}_2$ when $\mathbf{q} \in \mathcal{E}_K$.

When $\mathbf{q} \notin \mathcal{E}_K$, it holds that (99) $\leq \lambda \leq \mu_1$, and (100) ≥ 0 . Therefore, $GV_3(\mathbf{q}) \leq \mu_1$. \square

D.7 Proof of Lemma 4.12

LEMMA 4.11[RESTATED]. *Under Assumption 1 and Assumption 2, the probability $p_{\mathcal{B}}$ that an arriving job is blocked is bounded as*

$$p_{\mathcal{B}} \leq \frac{\tilde{d}_2}{\lambda} + \frac{52\tau_{1K}b^2}{\epsilon N}. \quad (4)$$

PROOF. Denote $B_\ell(\mathbf{q}) = \mathbb{1}\{\forall r \in N_L(\ell), q_r = b\}$. That is, whether all neighbors of port ℓ are full. Then by definition,

$$\begin{aligned} p_{\mathcal{B}} &= \frac{1}{\lambda_\Sigma} \sum_{\ell=1}^L \lambda_\ell \mathbb{E} [B_\ell(\bar{\mathbf{Q}})] \\ &= \frac{1}{\lambda_\Sigma} \sum_{\ell=1}^L \lambda_\ell \mathbb{E} \left[B_\ell(\bar{\mathbf{Q}}) \left| \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \leq 3 \right| \mathbb{P} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \leq 3 \right\} \right. \\ &\quad \left. + \frac{1}{\lambda_\Sigma} \sum_{\ell=1}^L \lambda_\ell \mathbb{E} \left[B_\ell(\bar{\mathbf{Q}}) \left| \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) > 3 \right| \mathbb{P} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) > 3 \right\} \right] \\ &\leq \frac{1}{\lambda_\Sigma} \sum_{\ell=1}^L \lambda_\ell \mathbb{E} \left[B_\ell(\bar{\mathbf{Q}}) \left| \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \leq 3 \right| \right] + \mathbb{P} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) > 3 \right\}. \end{aligned}$$

To bound $\mathbb{P} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) > 3 \right\}$, notice that $C^* \leq 1$, so

$$\mathbb{P} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) > 3 \right\} \leq \mathbb{P} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) > C^* + 2 \right\} \leq \frac{52\tau_{1K}b^2}{\epsilon N}$$

by Lemma 4.10.

Then for the case $\sum_{m=1}^K C_m(\mathbf{q}) \leq 3$, it holds that $\sum_{m=1}^K s_{m,b}(\mathbf{q}) \leq \frac{3}{b}$. Let \mathcal{I} be the set of servers of types no greater than K with queue length less than b . Then we know $|\mathcal{I}| \geq (1 - \frac{3}{b})N \geq \frac{\hat{\beta}}{2}N$ since $b \geq 6$. By Assumption 2, the total arrival rate of ports not connected with \mathcal{I} is thus upper bounded by $N\tilde{d}_2$. As a result,

$$p_{\mathcal{B}} \leq \frac{1}{\lambda_\Sigma} \sum_{\ell=1}^L \lambda_\ell \mathbb{E} \left[B_\ell(\bar{\mathbf{Q}}) \left| \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) \leq 3 \right| \right] + \mathbb{P} \left\{ \sum_{m=1}^K C_m(\bar{\mathbf{Q}}) > 3 \right\} \leq \frac{\tilde{d}_2}{\lambda} + \frac{52\tau_{1K}b^2}{\epsilon N}.$$

\square

D.8 Proof of Corollary 3.4

COROLLARY 3.4[RESTATED]. *Suppose that ϵ_N is both $o(1)$ and $\omega(N^{-0.5} \ln(N))$, and that both Assumptions 1 and 2 hold for G_N when N is sufficiently large. Then as $N \rightarrow \infty$, both JFSQ and JFIQ are asymptotically optimal, and the expected queueing delay converges to zero for both policies.*

PROOF. First since $\epsilon_N = \omega(\ln NN^{-0.5})$, there is always a b_N satisfying Assumption 1 when N is sufficiently large. Let $\bar{\mathbf{Q}}_N$ be the queue-length random variable, and let $p_{\mathcal{B}}^N$ be the blocking probability for the N -th system. Applying Theorem 3.2 gives

$$\mathbb{E} \left[\sum_{m=1}^M C_m(\bar{\mathbf{Q}}_N) \right] \leq C^* + \left(1 + \frac{\tau_{KM}}{2} \right) \epsilon_N + 2\sqrt{\frac{5\tau_{1M}b_N \ln N}{N}} + 60b_N^2 \sqrt{\frac{26\tau_{1K}\tau_{1M}}{\hat{\beta}_N \epsilon_N N}},$$

and $p_{\mathcal{B}}^N \leq \frac{\epsilon_N \mu_K}{2b_N \lambda} + \frac{52\tau_{1K} b_N^2}{\epsilon_N N}$ for N large enough.

Since $\epsilon_N = o(1)$, $\epsilon_N = \omega(N^{-0.5} \ln N)$, $\beta_N > \epsilon_N$ and b_N satisfies Assumption 1, it holds that $\lim_{N \rightarrow \infty} \mathbb{E} \left[\sum_{m=1}^M C_m(\bar{Q}_N) \right] = C^*$. Then by Little's Law, the expected mean response time $\mathbb{E}[T_N]$ of the N -th system is given by the mean number of jobs in the system divided by the effective arrival rate. Therefore,

$$\lim_{N \rightarrow \infty} \mathbb{E}[T_N] = \lim_{N \rightarrow \infty} \frac{\mathbb{E} \left[N \sum_{m=1}^M C_m(\bar{Q}_N) \right]}{\lambda_{\Sigma}(1 - p_{\mathcal{B}}^N)} \leq \frac{C^*}{\lambda \left(1 - \lim_{N \rightarrow \infty} \frac{\epsilon_N \mu_K}{2b_N \lambda} + \frac{52\tau_{1K} b_N^2}{\epsilon_N N} \right)} = \frac{C^*}{\lambda},$$

which matches the lower bound in Theorem 3.2. Therefore, JFSQ and JFIQ are asymptotically optimal in mean response time. On the other hand, let $\mathbb{E}[T_{\mathcal{W}}^N]$ be the expected waiting time of jobs, and let $\mathbb{E}[Z_N]$ be the expected service time in the N -th system. Then it holds $\mathbb{E}[T_N] = \mathbb{E}[T_{\mathcal{W}}^N] + \mathbb{E}[Z_N]$. Since $\mathbb{E}[Z_N] \geq \frac{C^*}{\lambda}$, $\mathbb{E}[T_{\mathcal{W}}^N] \geq 0$, and $\lim_{N \rightarrow \infty} \mathbb{E}[T_N] = \frac{C^*}{\lambda}$, it holds $\lim_{N \rightarrow \infty} \mathbb{E}[T_{\mathcal{W}}^N] = 0$. As a result, JFSQ and JFIQ obtain asymptotic zero queueing delays. \square

E PROOF OF RANDOM GRAPH RESULTS

Here we provide the missing proof of Theorem 3.6.

E.1 Proof of Theorem 3.6

THEOREM 3.6[RESTATE]. *Suppose that all ports share the same arrival rates, that is, $\lambda_{\ell} \equiv \bar{\lambda}$ for all $\ell \in \mathcal{L}$. Then following the same construction of graph G in Theorem 3.5 but with $H_j = 6 \left(-\ln p_j + \frac{\tilde{d}_j}{p_j \bar{\lambda}} \ln \frac{2\mu_1}{\tilde{d}_j} \right)$ for $j \in \{1, 2\}$, it holds that G satisfies Assumption 2 with probability at least $1 - 2 \binom{N}{N_{p_1}}^{-1}$. The total number of edges in G_N scales as $O \left(\frac{(N+L)b^3}{\epsilon} \ln \frac{b}{\epsilon} \right)$.*

PROOF. The proof is similar to that of Theorem 3.5. Let us follow the same notation in the proof of Theorem 3.5. Fix $j \in \{1, 2\}$. Similarly, let \mathcal{K} be any subset of \mathcal{L} satisfying $\sum_{\ell \in \mathcal{K}} \lambda_{\ell} > N\tilde{d}_j$, and \mathcal{I} be any subset of \mathcal{R}^j satisfying $|\mathcal{I}| \geq Np_j$. To bound $\mathbb{P}\{\mathcal{D}_{\mathcal{K}, \mathcal{I}}\}$, W.L.O.G., we can assume every port in \mathcal{K} has arrival rate less than $N\tilde{d}_j H_j$, otherwise $\mathbb{P}\{\mathcal{D}_{\mathcal{K}, \mathcal{I}}\} = 0$. Then following the same argument in the proof of Theorem 3.5, it holds $\mathbb{P}\{\mathcal{D}_{\mathcal{K}, \mathcal{I}}\} \leq \exp(-H_j N p_j)$.

The key step is to obtain a bound on the number of pairs of feasible \mathcal{K}, \mathcal{I} so that we can use the union bound. Let $N_{\mathcal{K}}^j, N_{\mathcal{I}}^j$ be the amount of such sets, respectively. W.L.O.G., assume that Np_j is an integer since $|\mathcal{I}|$ must be an integer. Also, as all ports share the same arrival rate $\bar{\lambda}$, we can assume $N\tilde{d}_j/\bar{\lambda}$ is an integer since the size of \mathcal{K} must exceed this value. Then it holds that

$$N_{\mathcal{K}}^j = \binom{L}{N\tilde{d}_j/\bar{\lambda}} \leq \binom{\lceil N\mu_1/\bar{\lambda} \rceil}{N\tilde{d}_j/\bar{\lambda}} \quad (103)$$

$$N_{\mathcal{I}}^j = \binom{N}{Np_j}. \quad (104)$$

We have the following lemma bounding a binomial number.

LEMMA E.1. *Fix an integer n . For any $0 < \alpha < \frac{1}{2}$, if αn is an integer, then $\ln \binom{n}{\alpha n} \leq -3\alpha n \ln \alpha$.*

PROOF. Let $k = \alpha n$. It holds that

$$\binom{n}{k} = \frac{n(n-1) \cdots (n-k+1)}{k!} \leq \frac{n^k}{k!}.$$

We know that $e^k = \sum_{i \geq 0} \frac{k^i}{i!}$. Therefore, $\frac{k^k}{k!} \leq e^k$. It then implies that

$$\binom{n}{k} \leq \frac{n^k}{k!} \leq \frac{e^k n^k}{k^k} = \left(\frac{en}{k}\right)^k.$$

As a result,

$$\ln \left(\binom{n}{\alpha n} \right) \leq \alpha n (1 - \ln(\alpha)) \leq -3n\alpha \ln \alpha$$

because $\alpha < \frac{1}{2}$. □

Now by the definition of p_j, \tilde{d}_j , it holds $p_j < \frac{1}{2}, \frac{N\tilde{d}_j/\bar{\lambda}}{[N\mu_1/\bar{\lambda}]} < \frac{1}{2}$. Then by Lemma E.1, when N is sufficiently large,

$$\ln \left(N_{\mathcal{K}}^j \right) \leq -3Np_j \ln p_j, \quad \ln \left(N_I^j \right) \leq -3N\tilde{d}_j/\bar{\lambda} \ln \left(\frac{2\mu_1}{\tilde{d}_j} \right). \quad (105)$$

Therefore, it holds that

$$\mathbb{P}\{C_j\} \leq N_{\mathcal{K}}^j N_I^j \exp(-H_j N p_j) \leq \exp \left(-Np_j H_j - 3Np_j \ln p_j - 3Np_j \frac{\tilde{d}_j}{p_j \bar{\lambda}} \ln \left(\frac{2\mu_1}{\tilde{d}_j} \right) \right). \quad (106)$$

By definition, $H_j = 6 \left(-\ln p_j - \frac{\tilde{d}_j}{p_j \bar{\lambda}} \ln \left(\frac{2\mu_1}{\tilde{d}_j} \right) \right)$. Then we can see

$$\mathbb{P}\{C_j\} \leq \exp(3Np_j \ln p_j) \leq \left(\frac{N}{Np_j} \right)^{-1}.$$

By the union bound, it holds that

$$\mathbb{P}\{C_1 \cup C_2\} \leq 2 \left(\frac{N}{Np_1} \right)^{-1}.$$

since $p_1 < p_2 < \frac{1}{2}$. Therefore, the probability that G_N satisfies Assumption 2 is at least $1 - 2 \left(\frac{N}{Np_1} \right)^{-1}$.

For the total number of edges used in G_N , consider the four types of connections on graph G_N as per Theorem 3.5 and Theorem 3.6 where we use different H_j . We bound the number of edges for each type as follows. First, through some calculations, $H_j = O \left(\left(1 + \frac{1}{b\bar{\lambda}} \right) \ln \left(\frac{b}{\epsilon} \right) \right)$, and $\frac{H_j}{\tilde{d}_j} = O \left(\frac{b^3 \bar{\lambda} + b^2}{\epsilon \bar{\lambda}} \ln \frac{b}{\epsilon} \right)$.

Then the number of ports with $\lambda_\ell \geq N \frac{\tilde{d}_1}{H_1}$ is bounded by $\frac{L\bar{\lambda}H_1}{N\tilde{d}_1} = O \left(\frac{(N+L)b^3}{N\epsilon} \ln \frac{b}{\epsilon} \right)$ because $\lambda_\Sigma = L\bar{\lambda}$. Therefore, the number of connections from them is bounded by $O \left(\frac{(N+L)b^3}{\epsilon} \ln \frac{b}{\epsilon} \right)$ since there are N servers. The same result holds for ports with $\lambda_\ell \geq N \frac{\tilde{d}_2}{H_2}$. Now for the remaining ports, the expected number of edges is upper bounded by

$$2 \sum_{\ell \in \mathcal{L}} \frac{\lambda_\ell}{N} \left(\frac{H_1}{\tilde{d}_1} + \frac{H_2}{\tilde{d}_2} \right) N = O \left(\frac{(N+L)b^3}{\epsilon} \ln \frac{b}{\epsilon} \right).$$

Then to sum up, the expected number of edges in G_N scales as $O \left(\frac{(N+L)b^3}{\epsilon} \ln \frac{b}{\epsilon} \right)$. □

F ADDITIONAL SIMULATION RESULTS

In this section, we provide missing details in the main text and give additional simulation results.

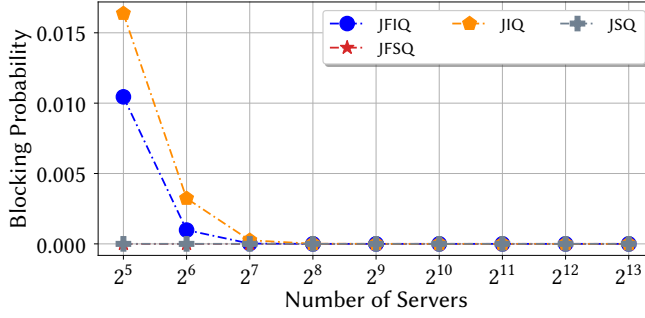


Fig. 4. The Blocking Probability of Different Routing Policies on Increasing-Sized Random Bipartite Graphs

F.1 Description of JSQ-(2,2)

In JSQ-(2,2)[19], there are two parameters p_F, p_S . Then for each arrival of jobs, we find a server as follows:

- (1) sample 2 fast servers and 2 slow servers;
- (2) if there is an idle fast server, route the job to this server;
- (3) if there is an idle slow server, route the job to this server with probability p_S , and route the job to the fast server with shorter queue with probability $1 - p_S$;
- (4) otherwise, route the job to the fast server with shorter queue with probability p_F ; and route the job to the slow server with shorter queue with probability p_S .

We set p_S, p_F to be the optimal values from Table 1 in [19].

F.2 Convergence of Blocking Probability

Fig. 4 provides the convergence of the blocking probability following the same setting as in Section 6.2. Unlike JSQ which is shown to be throughput optimal [11] (so is JFSQ), JIQ and JFIQ could lose the capacity of the system. As in Fig. 4, when we set the buffer size to be 5, the blocking probability of JIQ is around 1.5 percent, and that of JFIQ is around 1 percent. Interestingly, JFIQ seems to be more stable. Nevertheless, the blocking probability of both algorithms decreases swiftly as N increases.

F.3 Exploring More General Service Time Distribution

We present a preliminary study here that extends results proved in this paper. Roughly speaking, we consider the same setting as in Section 6.2. However, we allow the service time distribution to be hyper-exponential.

Still, suppose there are N servers in the system where N can scale up. Servers can be classified into four types with different service speed. Each type consists of the same amount of servers. Then let X be a hyper-exponential distribution such that $X \sim \text{Exp}(0.01)$ with probability 0.01, and $X \sim \text{Exp}(1)$ with probability 0.99. The coefficient of variation of X is around 7.071, which is higher than that of an exponential distribution. Then for a type i servers with $i \in \{1, 2, 3, 4\}$, we assume that the service time of a job at this server is independently and identically distributed as $2^{i-1}X$. Similarly, we can define the service rate of type- i servers as $\mu_i = \frac{1}{2^{i-1}\mathbb{E}[X]}$. Then the system load is defined as $\frac{4\lambda_\Sigma}{\sum_{i=1}^4 N\mu_i}$ where λ_Σ is the total arrival rate. We can also obtain the lower bound of the mean response time as in Proposition 3.1.

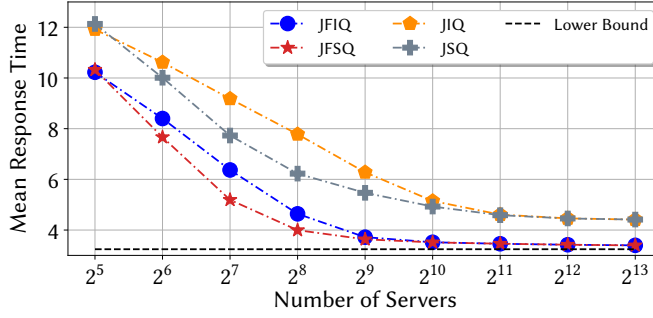


Fig. 5. The Mean Response Time of Different Routing Policies when Service Time is Hyper-Exponential

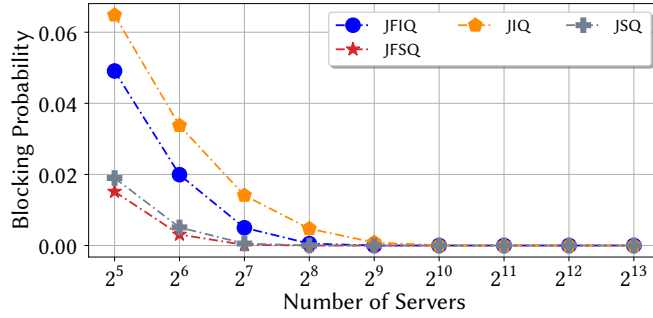


Fig. 6. The Blocking Probability of Different Routing Policies when Service Time is Hyper-Exponential

The buffer size is set as $b = 5$. Following the same setting of ports and construction of the random graph, we obtain Fig. 5 for the mean response time of different policies, and the blocking probability is shown in Fig. 6. Notice that the performance of each policy degrades a lot for small systems compared with Fig. 3. But when the system size scales up, both JFSQ and JFIQ have favorable mean response time, which is very close to the lower bound. It suggests that our theoretical results may hold for general distributions, which we leave for future studies.

Received August 2020; revised September 2020; accepted October 2020