2-2018

Mean-Field Analysis of Coding Versus Replication in Large Data Storage Systems

Bin Li
University of Rhode Island

Aditya Ramamoorthy
Iowa State University, adityar@iastate.edu

R. Srikant
University of Illinois at Urbana-Champaign

Mean-Field Analysis of Coding Versus Replication in Large Data Storage Systems

Abstract
We study cloud storage systems with a very large number of files stored in a very large number of servers. In such systems, files are either replicated or coded to ensure reliability, i.e., to guarantee file recovery from server failures. This redundancy in storage can further be exploited to improve system performance (mean file-access delay) through appropriate load-balancing (routing) schemes. However, it is unclear whether coding or replication is better from a system performance perspective since the corresponding queueing analysis of such systems is, in general, quite difficult except for the trivial case when the system load asymptotically tends to zero. Here, we study the more difficult case where the system load is not asymptotically zero. Using the fact that the system size is large, we obtain a mean-field limit for the steady-state distribution of the number of file access requests waiting at each server. We then use the mean-field limit to show that, for a given storage capacity per file, coding strictly outperforms replication at all traffic loads while improving reliability. Further, the factor by which the performance improves in the heavy traffic is at least as large as in the light-traffic case. Finally, we validate these results through extensive simulations.

Disciplines
Electrical and Computer Engineering | Systems and Communications

Mean-Field-Analysis of Coding versus Replication in Large Data Storage Systems

Bin Li, Aditya Ramamoorthy, and R. Srikant

Abstract

We study cloud-storage systems with a very large number of files stored in a very large number of servers. In such systems, files are either replicated or coded to ensure reliability, i.e., to guarantee file recovery from server failures. This redundancy in storage can further be exploited to improve system performance (mean file access delay) through appropriate load-balancing (routing) schemes. However, it is unclear whether coding or replication is better from a system performance perspective since the corresponding queueing analysis of such systems is, in general, quite difficult except for the trivial case when the system load asymptotically tends to zero. Here, we study the more difficult case where the system load is not asymptotically zero. Using the fact that the system size is large, we obtain a mean-field limit for the steady-state distribution of the number of file access requests waiting at each server. We then use the mean-field limit to show that, for a given storage capacity per file, coding strictly outperforms replication at all traffic loads while improving reliability. Further, the factor by which the performance improves in the heavy-traffic is at least as large as in the light-traffic case. Finally, we validate these results through extensive simulations.

I. Introduction

Data centers with a huge numbers of servers are used by many modern companies to serve their storage and computational needs. In this paper, we focus on the storage component of data centers. Consider a company like Facebook which stores a very large number of files, such as pictures, videos, etc., in a very large number of servers. Requests for downloading files arrive at the server, and the goal is to serve these requests with as little delay as possible. Additionally, for reliability purposes, each file is stored in multiple servers, using either simple replication or coding, to ensure that data is not lost even when some servers suffer from failures. The goal of this paper is to understand how this redundancy can be exploited to reduce the mean file access delay. In particular, we are interested in understanding whether coding always outperforms replication in terms of mean file access delay, under the same storage requirements.

To illustrate the difference between coding and replication, let us first consider the replication scheme. Suppose that each file is replicated in two servers, and assume that the time to download a file from a server is exponentially distributed with mean 1 and is independent across servers. Suppose that the load-balancing policy is to route an arriving request to server with the smallest queue length (i.e., the server with the smallest number of waiting requests). If the arrival rate of file download requests is very small, then the queue lengths (i.e., the number of requests awaiting service) at each server will be close to zero and therefore, an arriving request can be routed at random to any server containing the file. In this case, it is clear that the mean file access delay is just 1.

Next, let us consider the coding case. In particular, assume that the file is coded into 4 chunks, where the size of each chunk is half the size of the original file, and further the code is such that the file can be recovered from any two chunks. This can be achieved via Maximum Distance Separable (MDS) codes (e.g., [2]) with parameters $(4, 2)$, where the file is partitioned into two equal-size chunks A_1 and A_2, and the coded chunks A_1, A_2, $A_1 + A_2$ and $A_1 + 2A_2$ (the "+" operation is performed over an appropriate finite field) are stored in 4 different servers, respectively. Since each chunk is half the size of the original file, we assume that the amount of time required to download a chunk from a server is exponential with mean $1/2$. The natural load-balancing policy in this case is to choose the two least loaded of the four servers containing the file, and route an arriving request for the file to these two servers. Again, if the arrival rate of file download requests is close to zero, then all queue lengths will be close to zero and each arriving request can be routed to any two servers containing the file. Since we need both servers to complete serving the chunks that they contain, the mean file access delay is given by $\mathbb{E}[\max(X_1, X_2)]$, where X_1 and X_2 are i.i.d. exponential random variables with mean $1/2$. A straightforward calculation shows that this delay is equal to 0.75. Thus, it is quite clear that the mean file access delay is improved by 25% under coding compared with replication when the arrival rate is asymptotically negligible. However, it is unclear whether such a result extends to the case of non-zero request arrival rates. In such a case, queueing effects cannot be ignored. This poses significant challenges for the delay analysis. The

An earlier version of this paper has appeared in the Proceedings of IEEE International Conference on Computer Communications (INFOCOM), San Francisco, CA, USA, April 2016 [1].

Bin Li is with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: binli@uri.edu).

Aditya Ramamoorthy is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: aditya r@iastate.edu).

R. Srikant is with the Department of Electrical and Computer Engineering and Coordinate Science Lab, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail:rsrikant@illinois.edu).

main purpose of this paper is to address this open and difficult problem. Our contributions in this work can be summarized as follows:

• We first present a model of storage, routing, and file access in very large data centers. The interesting aspect of the model is that individual files become irrelevant, and the system can be viewed as a queueing model with a very large number of servers, thus facilitating the so-called mean-field analysis.

• Next, we carry out the mean-field analysis of the queueing system under both coding and replication, and derive their analytical expressions whose solutions yield the steady-state queue length distribution of each queue.

• Then, we utilize the mean-field-limit to show that coding strictly outperforms replication in terms of mean file access delay under the same storage requirements. We further characterize the improvement factor in the heavy-traffic regime, which is at least as large as that in the light-traffic regime. To the best of our knowledge, this is the first analytical result in the area of mean-field analysis that deals with the expected job delay rather than the expected task delay, where a job corresponds to a file access request containing a certain number of tasks (chunk downloading requests) depending on the coding scheme.

• Finally, we perform extensive simulations to validate our results, where we also study various service distributions, and more than one load-balancing scheme.

A. Related Work

Delay reduction via coding in cloud storage systems: The main goal of a cloud storage system is to provide high data reliability and fast file access. Recently, much work has gone into the design of algorithms that speed up the file access in cloud storage systems. For example, references (e.g., [3]–[5]) have performed simulation or testbed experiments to compare the delay performance of different coding schemes. Some other works have investigated the file access delay performance analytically. For example, the authors in [6] showed that the MDS code has a smaller mean file access delay than the simple file replication. In [7], the authors provided delay bounds under the MDS code. References (e.g., [5], [8]–[14]) studied the delay performance of redundant requests in various settings.

However, to the best of our knowledge, none of these works are able to characterize or analytically bound the performance improvement under coding compared with replication. In this paper, we work under the assumption that the underlying system is large and demonstrate rigorous lower bounds on the improvement in file access times due to coding.

Load-balancing in the large-system limit: A load-balancing algorithm distributes arriving jobs across servers with the goal of minimizing queueing delays. The analysis of load-balancing algorithms in any finite systems is quite challenging in general. References [15] and [16] first considered the celebrated power-of-d-choices ($d \geq 2$) load-balancing algorithm in the large-system limit, where each arriving job is forwarded to the shortest d randomly sampled queues. In such cases, any fixed number of queues become independent from each other and thus the delay characterization is tractable. There has been a considerable amount of recent work following the results in [15] and [16] studying various different load-balancing schemes with different amounts of overhead (e.g., [17]–[20]). More recently, Redundant Request with Killing (RRK) (e.g., [8], [12], [14], [21]) has received significant research attention, where each arriving job is replicated to d servers, and when any one of d jobs is processed, the rest of the jobs are killed. But, to the best of our knowledge, none of the previous papers have studied the joint performance of load balancing and storage schemes in the large-system limit.

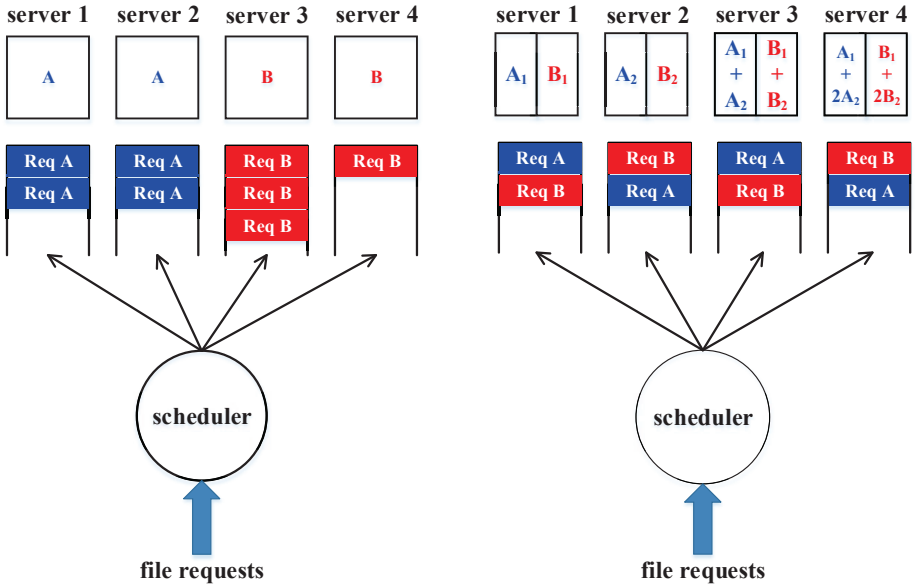
II. System Model

File storage scheme: We consider a cloud storage system with L servers, each of which stores a very large number of different types of files. Each file is stored using the Maximum Distance Separable (MDS) code with parameters (n, k) (see [2]), i.e., each file is encoded into n chunks with equal size stored at different servers, one for each server, and any k out of the n chunks are sufficient to recover the entire file. Since the storage space consumed at each server is $1/k$ of the size of the file, we assume that the time required for downloading data chunks are i.i.d. exponentially distributed[1] with mean of $1/k$. Note that the $(n, 1)$ code corresponds to the replication case, where each file is replicated at n different servers and thus we can download the desired file from any one of these n servers with exponential downloading time with mean 1.

Fig. 1(a) shows a small portion of the large storage system with $(2, 1)$ code, where file A is stored in servers 1 and 2, and file B is stored in servers 3 and 4. In order to download the file A, the scheduler can forward the file access request to either server 1 or server 2. Fig. 1(b) shows a part of the $(4, 2)$ coded system, where file A is divided into two equal-size halves A_1 and A_2, and the coded chunks A_1, A_2, $A_1 + A_2$, and $A_1 + 2A_2$ are stored in four different servers, respectively. In order to access file A, the scheduler needs to forward the file download request to any two of four servers. File A is obtained only when these two download requests are processed, i.e., when we receive two chunks of file A from two different servers.

Arrival process: Recall that each file is stored in n servers under the (n, k) code. Thus, there are a total of $\binom{L}{n}$ subsets of servers where a file could be stored. We assume that there are only $I = \Omega(L^2)$ files in the system and I is an increasing function of L. These I files are stored such that the load on each server is approximately the same. Thus, we can model the

[1]For the storage scheme with (n, k) code, the mean file access delay of the load-balancing scheme we consider is smaller under the positively correlated assumption than the i.i.d. assumption (cf. Section V). In this sense, we try to characterize the mean delay performance of a particular storage scheme in the worse scenario.



(a) $(2,1)$ code (b) $(4,2)$ code

Fig. 1: A small portion of a storage system. The letter inside the server box corresponds to the file it stores. Each server maintains a queue for download requests for the files it stores.

arrival process as follows: we assume that the arrival process of file download requests is Poisson with total arrival rate of $L\lambda$, where $\lambda \in (0,1)$. Further, each arrival requests a file uniformly at random from I files. Due to the property of the Poisson processes, this ensures that the load of any subset of servers of size n is independent with the same arrival rate.

Load-balancing algorithm: We assume that each server maintains a queue for file download requests that desire to download the chunks stored at the server, and processes these requests in the First-In-First-Out (FIFO) manner. Due to the MDS storage coding scheme, any k out of the n chunks are enough to obtain the entire file. Therefore, a natural load-balancing scheme is to forward an incoming file downloading request to the k least-loaded servers among n servers containing the file. In queueing theory jargon, upon job (file download request) arrival consisting of k tasks (data chunk retrieval request), forward these k tasks to the k least-loaded servers among n servers that can process this incoming job, one for each server. Each task processing time (chunk downloading time) follows exponential distribution with mean $1/k$. This load-balancing scheme is similar to the well-known *Batch Sampling* (BS) (e.g., [19], [22]). The main difference lies in that our considered load-balancing scheme uniformly selects one location containing n servers among $I = \Omega(L^2)$ rather than $\binom{L}{n}$ different locations upon each job arrival, where a location refers to a subset of servers with size of n and $\binom{L}{n}$ is much larger than I when $n > 2$. This is because there are only a total of I files in the cloud storage system. Nevertheless, we still refer our load-balancing scheme as Batch Sampling in the rest of the paper.

Another popular load balancing scheme that has attracted much attention recently is called Redundant Request with Killing (RRK) (e.g., [8], [12], [14], [21]). Under RRK, a request is sent to all servers where a file is stored, and when any k of these are served, the rest of the requests are killed. While this scheme is known to perform better than BS policy, it is under the assumption that the service time distributions are independent across servers. Later, in the simulations section, we show that the performance of RRK can be quite bad when service times across servers are correlated. For example, when a file is stored in equal-sized chunks across multiple servers, all requests for these chunks may have highly correlated service times. Thus, for our storage system, RRK has poor performance, so we do not study it here. On the other hand, in Section V, we will show that the performance of the BS scheme is worst under the assumption that the service times are independent across different servers. Hence, we study the system under this assumption in this paper.

Finally, we make a comment on the scenario that is being modeled in our paper and some of the other prior works (e.g., [5], [8], [9], [11]–[13]). Our work views the problem from the point of view of storage service provider. On the other hand, the previous works (e.g., [5], [8], [9], [11]–[13]) view the problem from the point of view of a customer who uses a cloud storage system. Thus, in these other works, the service time of a file is a complicated function of one's own file size, the storage server's speed and the service provided to other customers. Thus, their assumptions regarding service times can be quite different from ours.

Goal: It is quite obvious that coding can significantly improve system reliability compared with replication. In this paper, we would like to investigate whether coding also reduces file access delay under BS load-balancing algorithm. While we derive queue length distributions for general (n,k) codes, we mainly compare the mean file access delays of (nk,k) and $(n,1)$

(replication) codes[2], both of which have the same storage requirements, where $k \geq 2$. In particular, for the (nk, k) code, the file needs to be subdivided into k chunks, each of which is $1/k$-th the size of the original file. Coding is then applied on these k chunks to obtain nk coded chunks. Here, it is worth pointing out that none of existing works rigorously deal with the important and analytically hard problem of characterizing the mean job delay performance of the load-balancing schemes.

Let $\overline{W}^{(n,k)}$ be the mean file access delay under the (n, k) code. We first consider a trivial case, where the file request arrival rate is close to zero (also referred as the light-traffic regime). In such a case, queue lengths under both (nk, k) and $(n, 1)$ codes are close to zero and thus the queueing effect can be ignored. Therefore, it is obvious that $\overline{W}^{(n,1)} = 1$ under the replication scheme.

Under the (nk, k) code, we need to download k chunks from k different servers to recover the entire file, and thus

$$\overline{W}^{(nk,k)} = \mathbb{E}\left[\max_{i=1,2,\cdots,k} X_i\right],$$

where $X_i, \forall i$, are i.i.d. with exponential distribution with mean $1/k$. According to [23], we have

$$\overline{W}^{(nk,k)} = \frac{H(k)}{k},$$

where $H(m) \triangleq \sum_{l=1}^{m} 1/l$ denotes m^{th} harmonic number. Thus, the (nk, k) code reduces delay by $100(1 - H(k)/k)\%$ compared with the $(n, 1)$ code in the light-traffic regime. In order to get a sense of how much delay improvement in this case, we plot the delay improvement percentage $100(1 - H(k)/k)\%$ as a function of k. From Fig. 2, we can observe that the delay improvement is 25% when $k = 2$, 38.89% when $k = 3$, and the relative improvement becomes marginal as k further increases.

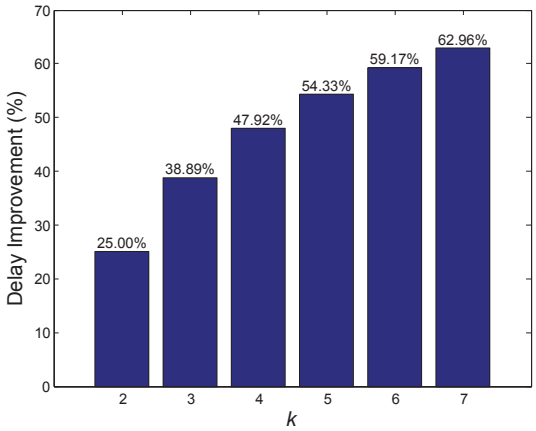


Fig. 2: Delay improvement under the (nk, k) code in the light-traffic regime (i.e., $\lambda \downarrow 0$)

This interesting observation raises the following two natural questions in the moderate and heavy traffic cases where the queueing effect cannot be ignored: (i) does the (nk, k) code always outperform the $(n, 1)$ code in terms of mean file access delay? (ii) if it does, then how much performance improvement can it achieve? The goal of this paper is to address these two open questions. In particular, we show that the (nk, k) code always outperforms the $(n, 1)$ code in terms of mean file access delay at all traffic loads, and the improvement factor in the heavy-traffic regime is at least as large as in the light-traffic regime.

III. Mean-Field Analysis

In this section, we will use mean-field analysis to study the mean file access delay performance under the (n, k) code. The underlying assumptions behind the mean-field analysis are validated in both Section IV and V.

Let $Q_l^{(L)}(t)$ be the length of the l^{th} queue at time t in a system with L queues. It is easy to check that the queue-length process $\{\mathbf{Q}^{(L)}(t)\}_{t\geq 0}$ is an irreducible and nonexplosive Markov chain. The following proposition further states that this Markov chain is positive recurrent and hence has a unique steady-state distribution.

Proposition 1: The Markov chain $\{\mathbf{Q}^{(L)}(t)\}_{t\geq 0}$ is positive recurrent. Moreover, the mean steady-state queue-length is finite, i.e.,

$$\mathbb{E}\left[\sum_{l=1}^{L} \widetilde{Q}_l^{(L)}\right] \leq \frac{L(1+\lambda)}{2(1-\lambda)}, \tag{1}$$

where $\widetilde{Q}_l^{(L)}$ is steady-state queue length of the l^{th} queue.

[2]If files are stored using $(n, 1)$ code such that arrival loads on each server are the same, then these files can also be stored using (nk, k) code to guarantee that arrival loads on each server are the same.

Proof: We first consider a quadratic Lyapunov function and study its conditional expected drift. Then, the desired result follows from the Foster-Lyapunov theorem. Please see Appendix A for details. ■

Due to the symmetry, all queues have the same steady-state distribution. Let $\{\pi_m^{(L)}\}_{m\geq 0}$ be the steady-state queue-length distribution of one queue, where $\pi_m^{(L)}$ denotes the probability that queue-length is exactly equal to m. Let $s_m^{(L)} \triangleq \sum_{j=m}^{\infty} \pi_m^{(L)}$ be the probability that queue-length is at least m. Note that $s_0^{(L)} = 1$ and $s_m^{(L)}$ is non-increasing with respect to m, i.e., $1 = s_0^{(L)} \geq s_1^{(L)} \geq s_2^{(L)} \geq \cdots \geq 0$. In addition, we have $\sum_{j=m}^{\infty} s_j^{(L)} < \infty, \forall m = 1, 2, \cdots$. Indeed, according to Proposition 1, we have

$$\sum_{j=m}^{\infty} s_j^{(L)} \leq \sum_{j=1}^{\infty} s_j^{(L)} = \mathbb{E}\left[\widetilde{Q}_l^{(L)}\right] < \infty, \quad \forall m \geq 1,$$

where we use the fact that $\mathbb{E}[Z] = \sum_{m=1}^{\infty} \Pr\{Z \geq m\}$ for any non-negative integer-valued random variable Z.

In this paper, our goal is to investigate the mean file access delay performance under the (n, k) code. In order to evaluate it accurately, it is important to obtain the queue-length distribution, i.e., the distribution of number of waiting download requests (queue-length) at each queue. However, queue lengths are correlated across queues and their distribution is hard to obtain in a system with finite number of queues. Fortunately, such correlations among queues become weaker and weaker as the number of servers increases. Indeed, as shown in Section IV, any fixed number of queues become independent of each other as the number of servers goes to infinity, i.e., $L \to \infty$, under a particular file storage manner. In such a case, the queue-length distribution can be exactly characterized. Such an analysis in the large-system limit is commonly referred as *mean-field analysis*. In addition, a cloud storage system typically contains a very large number of servers, and therefore the mean-field analysis is sufficiently accurate, as will be demonstrated in Section V via simulations.

A. Main Results

In this subsection, we present our main results on the mean file access delay under coding in the large-system limit (cf. Proposition 3). In particular, we characterize the delay improvement between (nk, k) and $(n, 1)$ codes, both of which have the same storage requirements.

Proposition 2: (i) The mean file access delay under the (nk, k) code is at least $(1 - H(k)/k)$ smaller than that under the $(n, 1)$ code for any arrival rate $\lambda \in (0, 1)$, i.e.,

$$\overline{W}^{(nk,k)} - \overline{W}^{(n,1)} \leq -\left(1 - \frac{H(k)}{k}\right). \tag{2}$$

(ii) In the light-traffic regime (i.e., $\lambda \downarrow 0$), the mean file access delay under the (nk, k) code improves $100\,(1 - H(k)/k)\,\%$ compared with the $(n, 1)$ code, i.e.,

$$\lim_{\lambda \downarrow 0} \frac{\overline{W}^{(nk,k)} - \overline{W}^{(n,1)}}{\overline{W}^{(n,1)}} = -\left(1 - \frac{H(k)}{k}\right). \tag{3}$$

In the heavy-traffic regime (i.e., $\lambda \uparrow 1$), the mean file access delay improvement under the (nk, k) code is at least $100\,(1 - H(k)/k)\,\%$ compared with the $(n, 1)$ code, i.e.,

$$\lim_{\lambda \uparrow 1} \frac{\overline{W}^{(nk,k)} - \overline{W}^{(n,1)}}{\overline{W}^{(n,1)}} \leq -\left(1 - \frac{H(k)}{k}\right). \tag{4}$$

The proof of Proposition 2 utilizes the steady-state queue-length distribution in the large-system limit (Section III-B) and the fact that the tail distribution of queue-length under the (nk, k) code decays at least as fast as that under the $(n, 1)$ code (see Lemma 3), and is available in Section III-C.

Our analysis shows that the (nk, k) code strictly outperforms the replication code at all traffic loads and its delay improvement in the heavy-traffic regime is at least as large as in the light-traffic regime. However, simulations in Section V indicate that the performance improvement in heavy-traffic is even better.

B. Steady-State Queue-Length Distribution

In this subsection, we obtain the queue-length distribution under the (n, k) code in the large-system limit, i.e., $L \to \infty$.

Recall that all queues have the same steady-state distribution because of symmetry. Let $\overline{Q}^{(n,k)}$ be a random variable with the same distribution as the steady-state distribution of the queue-length under the (n, k) code in the large-system limit. Let $\overline{\pi}_m \triangleq \Pr\{\overline{Q}^{(n,k)} = m\}$ be the steady-state probability that queue length is equal to m in the large-system limit, where $m = 0, 1, 2, \cdots$. Under the (n, k) code, whenever there is an arriving file access request, we forward these tasks to the k least-loaded servers among n servers containing the file, one for each server. Note that the time required for downloading the chunks are i.i.d. with exponential distribution with mean $1/k$. We assume that n servers containing the file requested by the

incoming job have independent queue-length distributions, as proved in Section IV. Note that the queue-length of each server increases or decreases at most by one. Each queue forms an independent Markov chain, as shown in Figure 3.

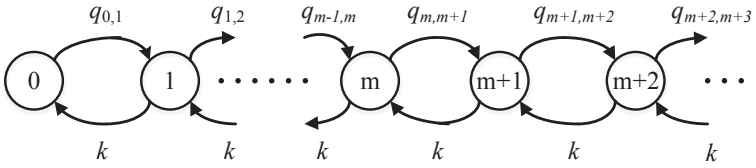


Fig. 3: The queue-length Markov chain of a single server in the large-system limit

According to the local balance equation, we have

$$\overline{\pi}_m q_{m,m+1} = k\overline{\pi}_{m+1}. \tag{5}$$

Therefore, in order to characterize the steady-state distribution $\{\overline{\pi}_m\}_{m\geq 0}$ in the large-system limit, we need to first obtain the transition rate $q_{m,m+1}$ when a file access request (job) arrives to a server with queue-length of m. Consider a particular server with queue-length of m. Its queue-length increases by 1 only when there is an arrival job and this server is one of the k least-loaded server among n servers containing the file that an incoming job requests. Note that $\overline{\pi}_m$ can also be interpreted as the fraction of servers with queue-length exactly equal to m in the large-system limit, which simply follows from the Strong Law of Large Numbers. Hence, $L\overline{\pi}_m$ is the average number of servers with queue-length of m and $L\overline{\pi}_m q_{m,m+1}\Delta$ is the average number of these servers that become of size $m+1$ due to an arrival in a small time interval Δ, which can also be represented as $L\lambda\Delta\sum_{i=1}^{k}\Pr\{\overline{Q}_{(i)}^{(n,k)} = m\}$. Thus, we have

$$\overline{\pi}_m q_{m,m+1} = \lambda\sum_{i=1}^{k}\Pr\{\overline{Q}_{(i)}^{(n,k)} = m\}, \tag{6}$$

where $\overline{Q}_{(i)}^{(n,k)}$ is the i^{th} smallest queue-length among n servers containing the file requested by the incoming job, i.e.,

$$\overline{Q}_{(1)}^{(n,k)} \leq \overline{Q}_{(2)}^{(n,k)} \leq \cdots \leq \overline{Q}_{(i)}^{(n,k)} \leq \cdots \leq \overline{Q}_{(n)}^{(n,k)}.$$

The next lemma gives the exact expression for $\sum_{i=1}^{k}\Pr\{\overline{Q}_{(i)}^{(n,k)} = m\}$. Let $\overline{s}_m \triangleq \sum_{j=m}^{\infty}\overline{\pi}_j$ denote the steady-state probability that queue-length is at least m in the large-system limit.

Lemma 1: The term $\sum_{i=1}^{k}\Pr\{\overline{Q}_{(i)}^{(n,k)} = m\}$ can be expressed as follows:

$$\sum_{i=1}^{k}\Pr\{\overline{Q}_{(i)}^{(n,k)} = m\} = f^{(n,k)}(\overline{s}_m) - f^{(n,k)}(\overline{s}_{m+1}), \tag{7}$$

where $f^{(n,k)}(x) \triangleq \sum_{l=1}^{k}\binom{n}{n-k+l}\binom{n-k+l-2}{l-1}(-1)^{l-1}x^{n-k+l}$, $x\in[0,1]$.

Proof: We first simplify the expression of $\Pr\{\overline{Q}_{(i)} \geq m\}$ by using the mean-field assumption, and then derive the expression for $\sum_{i=1}^{k}\Pr\{\overline{Q}_{(i)} \geq m\}$ through a little bit complicated algebraic operations. Please see Appendix B for details. ■

For example, $f^{(n,1)}(x) = x^n$ and $f^{(n,2)}(x) = nx^{n-1} - (n-2)x^n$. In general, the function $f^{(n,k)}(x)$ is quite complicated. However, it has several nice properties, which play an important role in later analysis.

Lemma 2: The function $f^{(n,k)}(x)$ (cf. Lemma 1) has the following properties:

(i) $f^{(n,k)}(x)$ is strictly increasing, differentiable and convex on the interval $[0,1]$;
(ii) $f^{(n,k)}(0) = 0$ and $f^{(n,k)}(1) = k$;
(iii) $f^{(n,k)}(x)$ has a bounded derivative, i.e.,

$$0 \leq \left(f^{(n,k)}(x)\right)' \leq n, \quad \forall x\in[0,1].$$

Proof: We consider the first and second derivatives of the function $f^{(n,k)}(x)$, and then utilize the subset-of-a-subset identity to get the desired result. Please see Appendix C for the proof. ■

Fig. 4 sketches the graph of the function $f^{(n,k)}(x)$. We are now ready to characterize the steady-state queue-length distribution in the large-system limit.

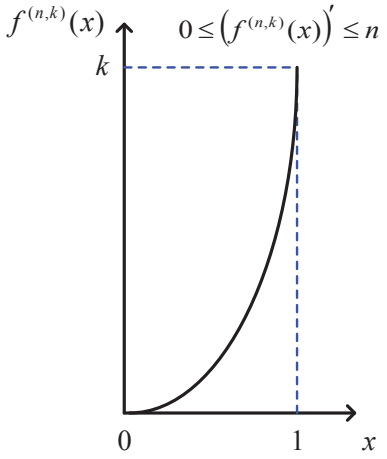


Fig. 4: The graph of the function $f^{(n,k)}(x)$

Proposition 3: The steady-state queue-length distribution of a single server under the (n,k) code in the large-system limit is unique and can be characterized as follows:

$$\begin{cases} \overline{s}_{m+1} = \lambda f^{(n,k)}(\overline{s}_m)/k & \text{for } m = 0,1,2,\cdots; \\ \overline{s}_0 = 1 & . \end{cases} \tag{8}$$

Proof: According to (5), (6) and Lemma 1, we have

$$\lambda\left(f^{(n,k)}(\overline{s}_m) - f^{(n,k)}(\overline{s}_{m+1})\right) = k(\overline{s}_{m+1} - \overline{s}_{m+2}), \tag{9}$$

for any $m = 0,1,2,\cdots$. Clearly if $\lambda f^{(n,k)}(\overline{s}_m)/k = \overline{s}_{m+1}$, then equation (9) holds. On the other hand, according to Lemma 2, the function $\lambda f^{(n,k)}(x)/k$ has a bounded derivative and thus it is Lipschitz. Also, $\lambda f^{(n,k)}(x)/k \in [0,1]$ since $f^{(n,k)}(x) \le k$ for all $x \in [0,1]$ and $\lambda \in (0,1)$. Therefore, the function $\lambda f^{(n,k)}(x)/k$ maps the convex and compact set $[0,1]$ to itself, and hence, according to the Schauder fixed point theorem, there exists a fixed point for the system of equations $\lambda f(\overline{s}_m)/k = \overline{s}_{m+1}, \forall m \ge 0$.

Next, we will show that this fixed point is unique. First, we note that

$$\overline{s}_{m+1} = \frac{\lambda}{k} f^{(n,k)}(\overline{s}_m) \overset{(a)}{\le} \lambda \overline{s}_m^{n/k} \overset{(b)}{\le} \lambda \overline{s}_m, \tag{10}$$

where step (a) utilizes the inequality $f^{(n,k)}(x) \le kx^{n/k}$ for any $x \ge 0$ (see Lemma 6 in Appendix D), and (b) is true since $n > k$ and $0 \le \overline{s}_m \le 1$. Inequality (10) directly implies $\sum_{j=m}^{\infty} \overline{s}_j < \infty$. Hence, we have $\sum_{j=m}^{\infty} f^{(n,k)}(\overline{s}_j) < \infty$. Indeed,

$$\sum_{j=m}^{\infty} f^{(n,k)}(\overline{s}_j) \overset{(a)}{=} \sum_{j=m}^{\infty} \left(f^{(n,k)}(z_j)\right)' \overline{s}_j \le n \sum_{j=m}^{\infty} \overline{s}_j < \infty,$$

where step (a) uses the fact that $f^{(n,k)}(\overline{s}_j) - f^{(n,k)}(0) = \left(f^{(n,k)}(z_j)\right)' \overline{s}_j$ for some $z_j \in [0, \overline{s}_j]$ according to the Mean-Value Theorem and the fact that $f^{(n,k)}(0) = 0$; (b) uses bounded derivative property of the function $f^{(n,k)}(x)$ (cf. Lemma 2). Therefore, by summing (9) over all $m \ge 0$, we obtain $\overline{s}_1 = \frac{\lambda}{k} f^{(n,k)}(\overline{s}_0)$. The uniqueness of the fixed point then follows from (9) by mathematical induction. ∎

Proposition 3 provides an iterative formula for exactly calculating the steady-state queue-length distribution under the (n,k) code. For example, under the $(n,1)$ code, i.e., power of n choices, according to Proposition 3, we have $\overline{s}_{m+1} = \lambda \overline{s}_m^n$ for all $m \ge 0$ and $\overline{s}_0 = 1$, which implies that $\overline{s}_m = \lambda^{\frac{n^m - 1}{n-1}}$. This exactly matches the results in [16] and [15]. Under the $(n,2)$ code, we have $\overline{s}_{m+1} = \frac{\lambda}{2}(n\overline{s}_m^{n-1} - (n-2)\overline{s}_m^n)$ for all $m \ge 0$ and $\overline{s}_0 = 1$ from the Proposition 3.

We are now ready to evaluate the mean file access delay.

C. Mean File Access Delay Analysis

In this subsection, we prove Proposition 2. We first show an important fact that the tail distribution of queue-length under the (nk,k) code decays at least as fast as that under the $(n,1)$ code, which implies that the average queue-length under the (nk,k) code is not greater than that under the $(n,1)$ code.

Lemma 3: The tail of queue-length distribution under the (nk, k) code decays at least as fast as that under the $(n, 1)$ code, i.e.,

$$\overline{s}_m \leq \hat{s}_m, \quad m = 0, 1, 2, \ldots, \tag{11}$$

where $\overline{s}_m$ and $\hat{s}_m$ denote the probability that the steady-state queue length is at least m under (nk, k) and $(n, 1)$ codes in the large-system limit, respectively.

The proof of Lemma 3 is available in Appendix D. Now, we are ready to show Proposition 2.

Proof of Proposition 2: Recall that under the (n, k) code, each job (file download request) contains k i.i.d. tasks (chunk download request) with exponential downloading time distribution with mean $1/k$. Upon job arrival, we forward its k tasks to the least-loaded k servers among n servers containing the file that the job request. Since a job is complete only when these k tasks are processed, if the queue lengths of these n servers are $\widehat{Q}_{(i)}^{(n,k)}$, $\forall i = 1, 2, \cdots, n$ when a job arrives, then this job experiences a delay equal to

$$\max_{i=1,2,\cdots,k} \sum_{j=1}^{\widehat{Q}_{(i)}^{(n,k)}+1} X_j^{(k,i)}, \tag{12}$$

where $X_j^{(k,i)}$, $\forall i, j$, are i.i.d. exponential random variables with mean $1/k$, and $\widehat{Q}_{(i)}^{(n,k)}$ is the i^{th} smallest queue-length among n servers seen by an incoming job, i.e., $\widehat{Q}_{(1)}^{(n,k)} \leq \widehat{Q}_{(2)}^{(n,k)} \leq \cdots \leq \widehat{Q}_{(n)}^{(n,k)}$.

Note that (12) is true since the remaining service time for the task in service is still exponential. We also note that $\widehat{Q}_{(i)}^{(n,k)}$, $\forall i = 1, 2, \cdots, n$ and $X_j^{(k,i)}, \forall i, j$, are independent. Therefore, the mean job delay $\overline{W}^{(n,k)}$ can be written as follows:

$$\overline{W}^{(n,k)} = \mathbb{E}\left[\max_{i=1,2,\cdots,k} \sum_{j=1}^{\widehat{Q}_{(i)}^{(n,k)}+1} X_j^{(k,i)}\right]. \tag{13}$$

Next, we compare the mean job delay under (nk, k) and $(n, 1)$ codes.

$$\begin{aligned}
\overline{W}^{(nk,k)} =& \mathbb{E}\left[\max_{i=1,2,\cdots,k} \sum_{j=1}^{\widehat{Q}_{(i)}^{(nk,k)}+1} X_j^{(k,i)}\right] \\
\overset{(a)}{\leq}& \mathbb{E}\left[\max\left\{\sum_{j=1}^{\widehat{Q}_{(1)}^{(nk,k)}+1} X_j^{(k,1)}, \max_{i=2,\cdots,k} \sum_{j=1}^{\widehat{Q}_{(1)}^{(nk,k)}+1} X_j^{(k,i)} + \max_{i=2,\cdots,k} \sum_{j=\widehat{Q}_{(1)}^{(nk,k)}+2}^{\widehat{Q}_{(i)}^{(nk,k)}+1} X_j^{(k,i)}\right\}\right] \\
\overset{(b)}{\leq}& \mathbb{E}\left[\max_{i=1,2,\cdots,k} \sum_{j=1}^{\widehat{Q}_{(1)}^{(nk,k)}+1} X_j^{(k,i)}\right] + \mathbb{E}\left[\max_{i=2,\cdots,k} \sum_{j=\widehat{Q}_{(1)}^{(nk,k)}+2}^{\widehat{Q}_{(i)}^{(nk,k)}+1} X_j^{(k,i)}\right],
\end{aligned} \tag{14}$$

where step (a) utilizes the fact that $\widehat{Q}_{(1)}^{(nk,k)} \leq \widehat{Q}_{(2)}^{(nk,k)} \leq \cdots \leq \widehat{Q}_{(k)}^{(nk,k)}$, and follows from the fact that $\max_i(x_i + y_i) \leq \max_i x_i + \max_i y_i$, for any non-negative real numbers x_i and y_i; (b) utilizes the fact that $\max\{x, y + z\} \leq \max\{x, y\} + z$, for any non-negative real numbers x, y and z. By repeating steps in deriving (14) on the term

$$\mathbb{E}\left[\max_{i=2,\cdots,k} \sum_{j=\widehat{Q}_{(1)}^{(nk,k)}+2}^{\widehat{Q}_{(i)}^{(nk,k)}+1} X_j^{(k,i)}\right],$$

we obtain

$$\begin{aligned}
\overline{W}^{(nk,k)} \leq& \mathbb{E}\left[\max_{i=1,2,\cdots,k} \sum_{j=1}^{\widehat{Q}_{(1)}^{(nk,k)}+1} X_j^{(k,i)}\right] + \sum_{l=2}^{k} \mathbb{E}\left[\max_{i=l,l+1,\cdots,k} \sum_{j=\widehat{Q}_{(l-1)}^{(nk,k)}+2}^{\widehat{Q}_{(l)}^{(nk,k)}+1} X_j^{(k,i)}\right] \\
\leq& \mathbb{E}\left[\sum_{j=1}^{\widehat{Q}_{(1)}^{(nk,k)}+1} \max_{i=1,2,\cdots,k} X_j^{(k,i)}\right] + \sum_{l=2}^{k} \mathbb{E}\left[\sum_{j=\widehat{Q}_{(l-1)}^{(nk,k)}+2}^{\widehat{Q}_{(l)}^{(nk,k)}+1} \max_{i=l,l+1,\cdots,k} X_j^{(k,i)}\right],
\end{aligned} \tag{15}$$

where the last step follows from the fact that

$$\max_{i=1,2,\cdots,a}\sum_{j=1}^{b}x_j^{(i)}\leq\sum_{j=1}^{b}\max_{i=1,2,\cdots,a}x_j^{(i)}$$

holds for any positive integers a, b, and non-negative real numbers $x_j^{(i)}$, $\forall i=1,2,\cdots,a,\forall j=1,2,\cdots,b$.

Since $X_j^{(k,i)}$ are i.i.d. exponential random variables, according to [23], we have

$$\mathbb{E}\left[\max_{i=1,2,\cdots,m}X_j^{(k,i)}\right]=\frac{1}{k}H(m), \tag{16}$$

where we recall that $H(m)\triangleq\sum_{i=1}^{m}1/i$ is the m^{th} harmonic number. Note that since $X_j^{(k,i)},i=l,l+1,\cdots,k$, are i.i.d. and independent from $\widehat{Q}_{(l)}^{(nk,k)}$, by utilizing (16), inequality (15) becomes

$$\begin{aligned}\overline{W}^{(nk,k)}\leq&\frac{1}{k}\left(\left(1+\mathbb{E}\left[\widehat{Q}_{(1)}^{(nk,k)}\right]\right)H(k)+\sum_{l=2}^{k}\left(\mathbb{E}\left[\widehat{Q}_{(l)}^{(nk,k)}\right]-\mathbb{E}\left[\widehat{Q}_{(l-1)}^{(nk,k)}\right]\right)H(k-l+1)\right)\\=&\frac{1}{k}\left(H(k)+\sum_{l=1}^{k}\frac{1}{k-l+1}\mathbb{E}\left[\overline{Q}_{(l)}^{(nk,k)}\right]\right),\end{aligned} \tag{17}$$

where we recall that $\overline{Q}_{(l)}^{(nk,k)}$ is the l^{th} smallest steady-state queue-length among nk servers, and the last step follows from PASTA property since the arrival process to any subset of queues of size nk is a Poisson process under the (nk,k) coding scheme.

On the other hand, the mean delay under the $(n,1)$ code can be written as follows:

$$\overline{W}^{(n,1)}=\mathbb{E}\left[\sum_{j=1}^{\widehat{Q}_{(1)}^{(n,1)}+1}X_j^{(1,1)}\right]\overset{(a)}{=}\mathbb{E}\left[\widehat{Q}_{(1)}^{(n,1)}\right]+1\overset{(b)}{=}\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]+1, \tag{18}$$

where step (a) follows from the fact that $\widehat{Q}_{(1)}^{(n,1)}$ and $X_j^{(1,1)},\forall j$, are independent; (b) follows from the PASTA property since the arrival process to any subset of queues of size n is a Poisson process under the $(n,1)$ coding scheme.

By using (17) and (18), we have

$$\overline{W}^{(nk,k)}-\overline{W}^{(n,1)}\leq-\left(1-\frac{H(k)}{k}\right)+\frac{1}{k}\sum_{l=1}^{k}\frac{1}{k-l+1}\mathbb{E}\left[\overline{Q}_{(l)}^{(nk,k)}\right]-\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]. \tag{19}$$

Note that

$$\frac{1}{k}\sum_{l=1}^{k}\frac{1}{k-l+1}\mathbb{E}\left[\overline{Q}_{(l)}^{(nk,k)}\right]\leq\frac{1}{k}\sum_{l=1}^{k}\mathbb{E}\left[\overline{Q}_{(l)}^{(nk,k)}\right]\leq\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right], \tag{20}$$

where the last step utilizes Lemma 4. By substituting (20) into (19), we have (2).

Lemma 4: The average queue-length of k shortest queues among nk servers under the (nk,k) code is not greater than the queue-length of the shortest queue among n servers under the $(n,1)$ code, i.e.,

$$\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}\left[\overline{Q}_{(i)}^{(nk,k)}\right]\leq\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]. \tag{21}$$

The proof of Lemma 4 directly follows from Lemma 3, and is available in Appendix E.

The mean job delay improvement under the (nk,k) code compared with the $(n,1)$ code in the light-traffic regime directly follows from the discussions in Section II. Next, we will investigate the mean job delay improvement in the heavy-traffic regime, i.e., $\lambda\uparrow 1$. According to (19), we have

$$\frac{\overline{W}^{(nk,k)}-\overline{W}^{(n,1)}}{\overline{W}^{(n,1)}}\leq-\left(1-\frac{1}{k}H(k)\right)+\frac{1}{k}\frac{\sum_{l=1}^{k}\frac{1}{k-l+1}\mathbb{E}\left[\overline{Q}_{(l)}^{(nk,k)}\right]-H(k)\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]}{1+\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]}, \tag{22}$$

which implies

$$\lim_{\lambda\uparrow 1}\frac{\overline{W}^{(nk,k)}-\overline{W}^{(n,1)}}{\overline{W}^{(n,1)}}\leq -\left(1-\frac{1}{k}H(k)\right)+\frac{1}{k}\lim_{\lambda\uparrow 1}\frac{\frac{\sum_{l=1}^{k}\frac{1}{k-l+1}\mathbb{E}\left[\overline{Q}_{(l)}^{(nk,k)}\right]}{\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]}-H(k)}{\frac{1}{\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]}+1}.$$

By utilizing Lemma 5, we have the desired result.

Lemma 5: (i) The mean queue-length of the shortest queue among n servers under the $(n,1)$ code in the heavy-traffic regime satisfies

$$\lim_{\lambda\uparrow 1}\frac{\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]}{-\log(1-\lambda)}=\frac{1}{\log n}; \tag{23}$$

(ii) The mean queue lengths of the k shortest queues among n servers under the (nk,k) code satisfy

$$\lim_{\lambda\uparrow 1}\frac{\sum_{i=1}^{k}\frac{1}{k-i+1}\mathbb{E}\left[\overline{Q}_{(i)}^{(nk,k)}\right]}{\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]}\leq H(k). \tag{24}$$

The proof of Lemma 5 is available in Appendix F.

IV. Proof of Asymptotic Independence

In this section, we justify the assumption on the asymptotic independence in the mean-field analysis that was used in the last section when the number of servers L is a prime number and the number of files I is equal to integer multiple of $L(L-1)/2$. In addition, these files are stored in a particular manner that is explained later. Simulation results in Section V indicate that our results hold even when L is not a prime number, I is the order of L^2, and files are uniformly randomly stored in different servers. The proof of asymptotic independence in such a general case remains an open question.

Under the (n,k) code, each file is stored in n different servers. We index the servers as $1,2,\ldots,L$. As is the case in practice, we assume $n\ll L$. In this section, we will call a sequence of the form a, $(a+d) \mod L$, $(a+2d) \mod L$,..., $(a+(n-1)d) \mod L$, an arithmetic progression with parameter d, where a can take values in $\{1,2,\ldots,L\}$. We assume that the files are stored in such a way that n servers containing the same file have indices that form an arithmetic progression with the common difference d in the circle with length L, where d takes values in $\{1,2,\ldots,(L-1)/2\}$. We call the set of n servers where a file is stored as the location of a file. Since L is a prime number, there are a total of L arithmetic progressions for a given parameter d and thus there are a total of $L(L-1)/2$ different locations to store files. Assuming that the number of files I is equal to an integer multiple of $L(L-1)/2$, then we just store these I files uniformly in these $L(L-1)/2$ locations.

Next, we will show that any fixed number of queues become independent of each other in the large-system limit under the above storage scheme, i.e., as the number of servers $L\to\infty$. To that end, we utilize the analytical approach in papers (e.g., [15], [19]), which basically includes three parts: (i) characterizing the differential equations governing the evolution of the system in the large-system limit and showing its convergence to the unique equilibrium point from any initial condition; (ii) proving that the Markov process of the number of jobs in the system converges to the differential equations over any finite time interval as $L\to\infty$; (iii) proving the interchange of limits, i.e., the convergence of the stationary probability measure of the Markovian system to the Dirac measure of unique equilibrium point is independent of the order of the large-system (i.e., $L\to\infty$) and infinite-time limits (i.e., $t\to\infty$). It is worth noting that the asymptotic independence results were established under the batch-sampling in [19] when the batch-size (i.e., the number of tasks of each job) goes to infinity and number of files is $I=\binom{L}{n}$. However, the batch-size in our system is always finite and I is just integer multiple of $L(L-1)/2$, and thus the result in [19] do not directly apply.

First, we characterize the differential equations governing the system dynamics under the (n,k) code in the large-system.

$$\frac{ds_m(t)}{dt}=\lambda\left(f^{(n,k)}(s_{m-1}(t))-f^{(n,k)}(s_m(t))\right)-k\left(s_m(t)-s_{m+1}(t)\right),\forall m=1,2,\cdots, \tag{25}$$

where $f^{(n,k)}(x)$ is defined in Lemma 1, the initial condition $\mathbf{s}(0)\triangleq\{\eta_m\}_{m\geq 0}\in\mathcal{S}$ and $\mathcal{S}\triangleq\{\boldsymbol{\zeta}\triangleq(\zeta_m)_{m\geq 0}:1=\zeta_0\geq\zeta_1\geq\cdots\geq 0$ and $\sum_{m=1}^{\infty}\zeta_m<\infty\}$. We further equip the space $\mathcal{S}$ with the metric

$$d(\mathbf{x},\mathbf{y})=\sup_{m\geq 1}\frac{|x_m-y_m|}{m}, \tag{26}$$

where $\mathbf{x}=(x_m)_{m\geq 0}$ and $\mathbf{y}=(y_m)_{m\geq 0}$ are within the space $\mathcal{S}$. Then, it is easy to check that the space $\mathcal{S}$ is compact.

Differential equations (25) are derived from the Markov chain in Figure 3. Regard $s_m(t)$ as the fraction of queues with length at least m at time t. The terms of the first group in (25) correspond to the queue length increase due to an arriving file access request (job). Note that the queue-length of a queue with length of m increases by 1 only when there is a job arrival

and the considered queue is one of the k shortest queues among n servers containing the requested file. This combined with Lemma 1 implies that the probability that the queue-length increases by 1 is equal to $\lambda(f^{(n,k)}(s_{m-1}(t)) - f^{(n,k)}(s_m(t)))$. The terms of the second group in (25) is simply because of a departure with rate k. Also, we note that the initial condition $\mathbf{s}(0) = (\eta_m)_{m\geq 0} \in \mathcal{S}$ satisfies $\sum_{m=1}^{\infty} \eta_m < \infty$, which is related to the average queue-length at one server. Therefore, the condition $\mathbf{s}(0) \in \mathcal{S}$ simply requires that the average queue-length per server is bounded initially.

Let $F_m^{(n,k)}(\mathbf{s}(t)) \triangleq \lambda\left(f^{(n,k)}(s_{m-1}(t)) - f^{(n,k)}(s_m(t))\right) - k\left(s_m(t) - s_{m+1}(t)\right), \forall m = 1, 2, \cdots$, denote the drift of $s_m(t)$ at point $\mathbf{s}(t)$. Then, the system of the above differential equations (25) can be rewritten as

$$\frac{d\mathbf{s}(t)}{dt} = \mathbf{F}^{(n,k)}(\mathbf{s}(t)), \tag{27}$$

where $\mathbf{F}^{(n,k)}(\mathbf{s}) = (F_1^{(n,k)}(\mathbf{s}), F_2^{(n,k)}(\mathbf{s}), \cdots, F_m^{(n,k)}(\mathbf{s}), \cdots)$. The *equilibrium point* of (25) is obtained by setting $\mathbf{F}^{(n,k)}(\mathbf{s}(t)) = 0$. It is easy to verify that the steady-state distribution $\overline{\mathbf{s}} = \{\overline{s}_m\}_{m\geq 0}$ (cf. Proposition 3) in the large-system limit is the unique equilibrium point.

Next, we show that the solution of (25) with any initial condition within $\mathcal{S}$ converges to the unique equilibrium point.

Proposition 4: For any initial condition $\boldsymbol{\eta} \in \mathcal{S}$, the solution $\mathbf{s}(t)$ of differential equations (25) converges to the unique equilibrium point $\overline{\mathbf{s}}$.

Here, we want to point out that the Lyapunov function used in [16] and [19] does not work in our system. We follow the approach in [15, Section 3] by utilizing the monotone property of the function $f^{(n,k)}(x)$ (cf. Lemma 2). The detailed proof is available in Appendix H.

Let $\Phi_m^{(L)}(t)$ be the number of queues with length at least m, and $\phi_m^{(L)}(t) \triangleq \Phi_m^{(L)}(t)/L$ be the fraction of queues with length at least m in the system with L servers. The following theorem states that $\boldsymbol{\phi}^{(L)}(t) \triangleq (\phi_1^{(L)}(t), \phi_2^{(L)}(t), \cdots, \phi_m^{(L)}(t), \cdots)$ converges to $\mathbf{s}(t)$ for any bounded time interval $[0, T]$ as $L \to \infty$, where $T \geq 0$ is some finite constant.

Proposition 5: Assume that $\boldsymbol{\phi}^{(L)}(0) \to \mathbf{s}(0)$ in probability, where $\mathbf{s}(0)$ is a deterministic condition with $\mathbf{s}(0) \in \mathcal{S}$. Then, we have

$$\lim_{L\to\infty} \sup_{0\leq t\leq T} \|\boldsymbol{\phi}^{(L)}(t) - \mathbf{s}(t)\|_1 = 0 \text{ in probability}, \tag{28}$$

where $\|\mathbf{x}\|_1 \triangleq \sum_{m=0}^{\infty} |x_m|$ is the l_1 norm of the vector $\mathbf{x} \in \mathcal{S}$.

Here, we consider an infinite-dimensional state space, where the classical Kurtz's Theorem (e.g., [24]) only considers the finite-dimensional state space. To that end, we carefully partition the state space into two subspaces, where one of them is finite-dimensional. In this sense, our proof is regarded as an extension of the Kurtz's Theorem. We note that [19] uses a similar idea to show the Kurtz's result under the batch-sampling when the batch-size goes to infinity and the number of files I is exactly equal to $\binom{L}{n}$, while we establish a similar result for the case with any finite batch-size and I equal to the integer multiple of $L(L-1)/2$. The proof is available in Appendix I.

Having established Propositions 4 and 5, by the Interchange of Limits Theorem (i.e., the convergence of the probability measure of the finite system to the Dirac measure of the unique equilibrium point is independent of the order of the large-system and infinite-time limits) in [19, Theorem 9] and showing that the steady-state distribution of any fixed number of queues in the large-system limit has a product form, we have the following result (also see [19, Corollary 12]), which establishes the asymptotic independence among any fixed number of queues in the large-system limit.

Proposition 6: Consider a set of l queues, and without loss of generality, assume the queues are $1, 2, \cdots, l$. Let $\pi^{(L)}(Q_1, Q_2, \cdots, Q_l)$ denote the steady-state queue-length distribution of these l queues in the system with L queues. Then, these queues become identically and independently distributed with distribution $\overline{\mathbf{s}}$ in the large-system limit, i.e.,

$$\lim_{L\to\infty} \pi^{(L)}(Q_1, Q_2, \cdots, Q_l) = \prod_{i=1}^{l} \overline{s}_{Q_i}. \tag{29}$$

V. Simulation Results

In this section, we provide simulation results to compare the mean file access delay performance between coding and replication in the system with $L = 1,000$ servers and $I = 1,000,000$ files. In particular, we first verify the accuracy of the mean-field analysis and then investigate the delay improvement under coding. Then, we evaluate the impact of correlation of the chunk downloading time on the mean delay performance for two different load-balancing algorithms.

A. Validation of the Mean-Field Analysis

In this subsection, we first validate the accuracy of the mean-field analysis, and then illustrate the differences in mean file access delay performance between coding and replication, where we assume that the chunk downloading time follows exponential distribution. Given the queue-length distribution (cf. Proposition 3), we are able to calculate the mean file access delay under the (n, k) code according to (13) through Monte Carlo methods. In particular, at each time slot, generate n i.i.d.

queue-length random variables according to its steady-state probability distribution in the large-system limit (cf. Proposition 3), then pick k smallest ones and calculate the delay through (13). Then, the time-average delay can be regarded as the mean delay. The markers in Fig. 5 (corresponding to theoretical results) were obtained in this manner, whereas the simulation results were obtained via an event-driven simulation of the whole system.

From Fig. 5, we first observe that the simulation results match the theoretical results very well under different coding schemes, which validates the accuracy of the mean-field analysis in the system with a large number of servers. In addition, Fig. 5 shows the mean file access delay performance under the (nk, k) code, where $k = 1, 2, 3, 4, 5$. Recall that $k = 1$ corresponds to the replication code. We can see from Fig. 5 that the mean file access delay performance improves as k increases, where the delay improvement is most significant from $k = 1$ to $k = 2$. This is also expected from our theoretical analysis. In addition, for a fixed storage coding scheme, its delay improvement compared with the replication code increases as the arrival rate λ increases. We also consider the case with i.i.d. chunk downloading time with distribution the same as $1/(2k) + \text{Exp}(2k)$, where $\text{Exp}(2k)$ is exponential distributed with mean $1/(2k)$. This downloading time distribution was used in [12] to model the data downloading time in Amazon AWS system. The simulation results are shown in Fig. 6, where we have similar observations with the case with exponential downloading time (cf. Fig. 5).

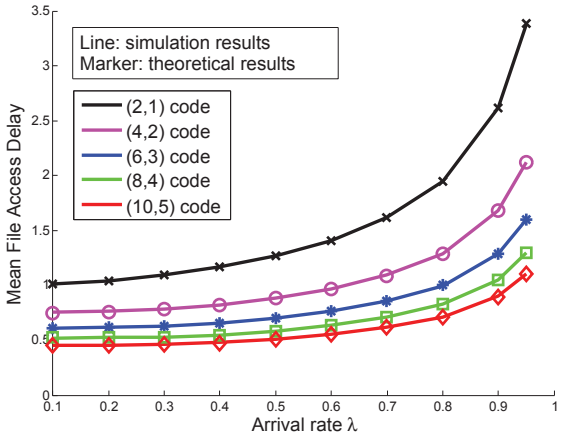


Fig. 5: Exp downloading time

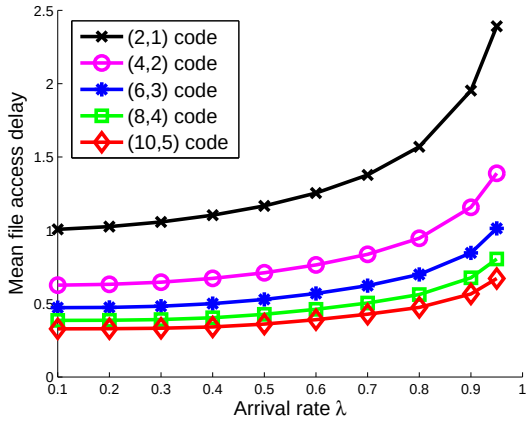
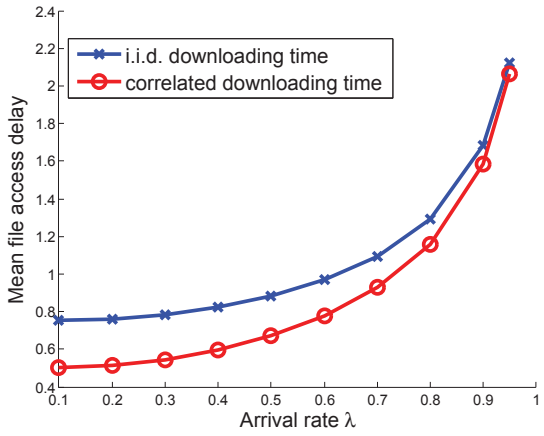
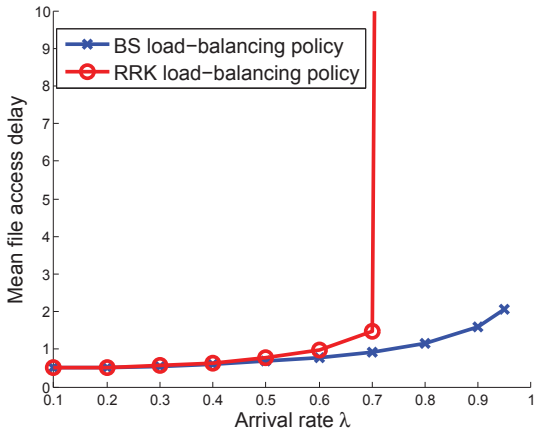


Fig. 6: Constant+Exp downloading time



(a) BS load-balancing scheme

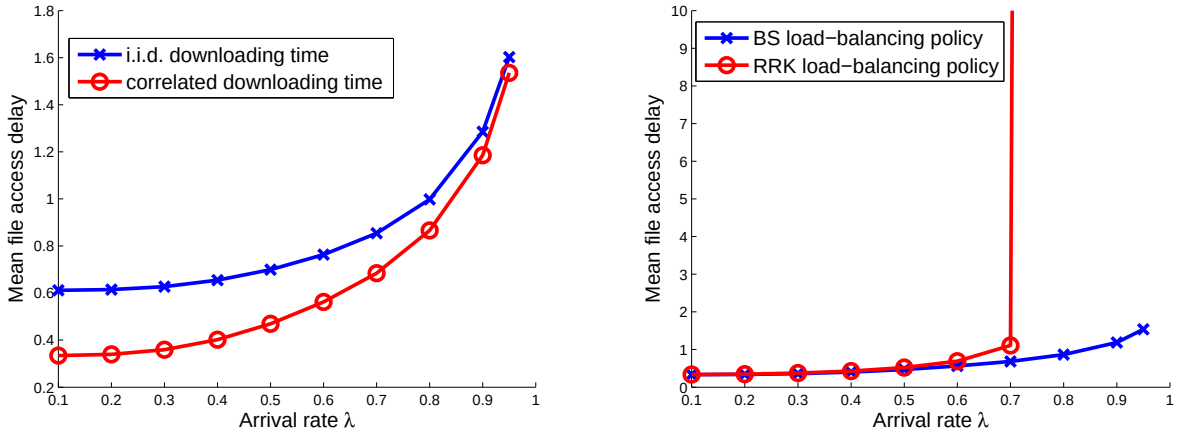


(b) Correlated downloading time

Fig. 7: Impact of correlated downloading time on the delay performance of $(4, 2)$ code

B. Impact of Correlated Downloading Time Distribution

In this subsection, we consider another popular load-balancing scheme, called Redundant Request with Killing (RRK), under the storage scheme with (n, k) code. Recall that under the RRK load-balancing scheme, upon a file access request arrival, it forwards n requests to n servers containing the file and the entire file is obtained once k out of n downloading requests are processed.



(a) BS load-balancing scheme

(b) Correlated downloading time

Fig. 8: Impact of correlated downloading time on the delay performance of $(6, 3)$ code

Here, we consider both i.i.d. and correlated downloading time cases. In the case with i.i.d. downloading time, the time required for downloading data chunks are i.i.d. with exponential with mean $1/k$. In the case with correlated downloading time, the time required for downloading chunks associated with a file are exactly the same and follows exponential distribution with mean $1/k$.

Fig. 7 studies the impact of correlations on delay performance of the Batch Sampling (BS) and RRK load-balancing algorithms under the $(4, 2)$ storage scheme. From Fig. 7(a), we can observe that for the BS load-balancing policy, the mean delay under the correlated downloading time is always better than that under the i.i.d. downloading time, with larger improvement in the lower traffic regime. In this sense, the correlation of the chunk downloading time actually helps improve the delay performance of the BS policy. Thus, the results in the paper may be interpreted as characterizing the worst-case performance of the BS policy. However, from Fig. 7(b), we can see that this correlation significantly degrades the system performance of the RRK algorithm especially when the traffic load is high. We have the same observations from Fig. 8 that shows the simulation results for the storage scheme with $(6, 3)$ code. Thus, the efficiency of the RRK policy heavily depends on the independence assumption on the chunk downloading time as we discussed in Section II. For this reason, we only analytically study the BS policy in this paper.

VI. Conclusions

In this paper, we studied the mean file access delay performance under coding in cloud storage systems with a very large number of files stored in a very large number of servers. We formulated an appropriate load-balancing problem, and studied its delay performance in the large-system limit, i.e., when the number of servers goes to infinity. In particular, we obtained the steady-state distribution of the number of file access requests waiting at each server, and utilized this to show that coding always improves the mean file access delay compared with the simple replication scheme at all traffic loads, without sacrificing any storage and reliability. We further show that the improvement factor by coding in the heavy-traffic regime is at least as large as in the light-traffic regime. Finally, extensive simulations are performed to validate our theoretical results.

Appendix A
Proof of Proposition 1

We ignore the superscript (L) of $Q_l^{(L)}(t)$ for simplicity. Consider the Quadratic Lyapunov function $V(\mathbf{Q}(t)) = \sum_{l=1}^{L} Q_l^2(t)$. Let $\mathbf{x}, \mathbf{y}$ be the state of the underlying Markov chain, and $q_{\mathbf{x},\mathbf{y}}$ denote the transition rate from state $\mathbf{x}$ to state $\mathbf{y}$. According to the Foster-Lyapunov theorem (see [25, Theorem 9.1.8]) for continuous-time Markov chain, we consider its Lyapunov drift as follows:

$$\begin{aligned}
&\sum_{\mathbf{y}\neq\mathbf{x}} q_{\mathbf{x},\mathbf{y}} \left(V(\mathbf{y}) - V(\mathbf{x})\right) \\
=&\sum_{l=1}^{L} q_{\mathbf{x},\mathbf{x}-\mathbf{e}_l} \left(V\left((\mathbf{x}-\mathbf{e}_l)^+\right) - V(\mathbf{x})\right) + \sum_{\mathbf{y}\in\Theta_{\mathbf{x}}} q_{\mathbf{x},\mathbf{y}} \left(V\left(\mathbf{y}\right) - V(\mathbf{x})\right), \qquad (30)
\end{aligned}$$

where the last step is true for $z^+ \triangleq \max\{z, 0\}$, $\mathbf{e}_l$ being a $L \times 1$ vector such that $\mathbf{e}_l[l] = 1$ and $\mathbf{e}_l[l'] = 0$ for any $l' \neq l$, and $\Theta_{\mathbf{x}}$ being the set of possible states of the Markov chain that can be reached from the state $\mathbf{x}$ when there is a job arrival (file access request).

For the term $\sum_{l=1}^{L} q_{\mathbf{x},\mathbf{x}-\mathbf{e}_l} \left(V\left((\mathbf{x}-\mathbf{e}_l)^+\right) - V(\mathbf{x})\right)$, we have

$$\begin{aligned}
&\sum_{l=1}^{L} q_{\mathbf{x},\mathbf{x}-\mathbf{e}_l} \left(V\left((\mathbf{x}-\mathbf{e}_l)^+\right) - V(\mathbf{x})\right) \\
\overset{(a)}{\leq} &\sum_{l=1}^{L} k\left((x_l-1)^2 - x_l^2\right) \\
= &-2k\sum_{l=1}^{L} x_l + kL,
\end{aligned} \tag{31}$$

where the step (a) uses the fact that the departure rate of each task (chunk download request) at each queue is k, and the fact that $\left(z^+\right)^2 \leq z^2$ for any real number z.

For the term $\sum_{\mathbf{y}\in\Theta_{\mathbf{x}}} q_{\mathbf{x},\mathbf{y}} \left(V(\mathbf{y}) - V(\mathbf{x})\right)$, we have

$$\begin{aligned}
&\sum_{\mathbf{y}\in\Theta_{\mathbf{x}}} q_{\mathbf{x},\mathbf{y}} \left(V(\mathbf{y}) - V(\mathbf{x})\right) \\
\leq &L\lambda \sum_{l=1}^{L} \left((x_l+1)^2 - x_l^2\right) \times \frac{k}{L} \\
= &2k\lambda \sum_{l=1}^{L} x_l + kL\lambda,
\end{aligned} \tag{32}$$

where the first step is established by comparing the batch-sampling with the randomized load-balancing policy that forwards k tasks to randomly selected k queues with replacement, one for each queue. Indeed, conditioned on the n sampled queues, e.g., $Q_1 \leq Q_2 \leq \cdots \leq Q_n$, the Lyapunov drift can be represented as

$$\begin{aligned}
&\mathbb{E}\left[\sum_{l=1}^{n} \left((Q_l + a_l)^2 - Q_l^2\right) \middle| (Q_1, Q_2, ..., Q_n)\right] \\
= &2\mathbb{E}\left[\sum_{l=1}^{n} Q_l a_l \middle| (Q_1, Q_2, ..., Q_n)\right] + k,
\end{aligned} \tag{33}$$

whenever there is an arrival event under any load-balancing policy, where $\sum_{l=1}^{n} a_l = k$ and $a_l \in \{0, 1\}$. It is easy to see that

$$\mathbb{E}\left[\sum_{l=1}^{n} Q_l a_l \middle| (Q_1, Q_2, ..., Q_n)\right] = \sum_{l=1}^{k} Q_k, \tag{34}$$

under our considered load-balancing scheme. Under the above randomized load-balancing policy,

$$\mathbb{E}\left[Q_l a_l | (Q_1, Q_2, \ldots, Q_n)\right] = \frac{k}{L} Q_l, \forall l = 1, 2, ..., n. \tag{35}$$

On the other hand,

$$\mathbb{E}\left[Q_l a_l\right] = \sum_{\mathbf{G}\in\mathcal{G}} \mathbb{E}\left[Q_l a_l | \mathbf{G}\right] \Pr\{\mathbf{G}\}, \tag{36}$$

where $\mathcal{G}$ = the set of n sampled servers containing server l. Note that each server contains In/L files and each file is stored in $n-$tuple of servers, where we recall that I is the number of files in the system. This implies that each server belongs to In/L $n-$tuple of servers and thus $|\mathcal{G}| = In/L$.

Also, due to the symmetry, $\mathbb{E}\left[Q_l a_l | \mathbf{G}\right], \forall \mathbf{G} \in \mathcal{G}$, have the same value. Therefore, combining equations (35), (36), and the fact that $|\mathcal{G}| = In/L$ and $\Pr\{\mathbf{G}\} = 1/I, \forall \mathbf{G} \in \mathcal{G}$, we have $\mathbb{E}\left[Q_l a_l | (Q_1, Q_2, ..., Q_n)\right] = \frac{k}{n} Q_l, \forall l = 1, 2, ..., n$, which implies that

$$\mathbb{E}\left[\sum_{l=1}^{n} Q_l a_l \middle| (Q_1, Q_2, ..., Q_n)\right] = \frac{k}{n} \sum_{l=1}^{n} Q_l, \tag{37}$$

under the randomized load-balancing policy. By using the assumption that $Q_1 \leq Q_2 \leq \cdots \leq Q_n$, we have

$$\begin{aligned}\frac{k}{n}\sum_{l=1}^{n} Q_l \geq & \frac{k}{n}\left(\sum_{l=1}^{k} Q_l + (n-k)Q_{k+1}\right) \\ \geq & \frac{k}{n}\left(\sum_{l=1}^{k} Q_l + \frac{n-k}{k}\sum_{l=1}^{k} Q_l\right) = \sum_{l=1}^{k} Q_l. \end{aligned} \tag{38}$$

This implies that, conditioned on n sampled queues, the Lyapunov drift upon an arrival under our considered load-balancing scheme is not greater than that under the above mentioned randomized load-balancing policy.

Therefore, we have

$$\begin{aligned}&\sum_{\mathbf{y}\neq\mathbf{x}} q_{\mathbf{x},\mathbf{y}}\left(V(\mathbf{y}) - V(\mathbf{x})\right) \\ \leq & -2k(1-\lambda)\sum_{l=1}^{L} x_l + kL(1+\lambda). \end{aligned} \tag{39}$$

Since $\lambda \in (0,1)$, according to the Foster-Lyapunov theorem (see [25, Theorem 9.1.8]), the underlying Markov chain is positive recurrent, and hence its steady-state distribution exists. Then, (1) follows from [26, Proposition 2.2.3].

APPENDIX B
PROOF OF LEMMA 1

In the rest of proof, we omit the superscript (n,k) of $\overline{Q}_{(i)}^{(n,k)}, \forall i = 1, 2, \cdots, n$, for simplicity. Since there are n sampled queues with $\overline{Q}_{(1)} \leq \overline{Q}_{(2)} \leq \cdots \leq \overline{Q}_{(n)}$, we have

$$\begin{aligned}&\Pr\{\overline{Q}_{(i)} \geq m\} \\ \overset{(a)}{=} &\Pr\{n-i+1 \text{ or more of } \overline{Q}_l\text{'s are} \geq m\} \\ \overset{(b)}{=} &\sum_{j=n-i+1}^{n}\binom{n}{j}\overline{s}_m^j(1-\overline{s}_m)^{n-j} \\ \overset{(c)}{=} &\sum_{j=n-i+1}^{n}\binom{n}{j}\overline{s}_m^j\sum_{l=0}^{n-j}\binom{n-j}{l}(-\overline{s}_m)^l \\ = &\sum_{j=n-i+1}^{n}\sum_{l=0}^{n-j}\binom{n}{j}\binom{n-j}{l}(-1)^l\ \overline{s}_m^{j+l} \\ \overset{(d)}{=} &\sum_{d=n-i+1}^{n}\overline{s}_m^d\sum_{j=n-i+1}^{d}\binom{n}{j}\binom{n-j}{d-j}(-1)^{d-j} \\ \overset{(e)}{=} &\sum_{d=n-i+1}^{n}\binom{n}{d}\overline{s}_m^d\sum_{j=n-i+1}^{d}\binom{d}{j}(-1)^{d-j}, \end{aligned} \tag{40}$$

where the step (a) follows from the fact that the i^{th} order statistic is greater than or equal to m if and only if there are $n-i+1$ or more of $\overline{Q}_l$'s that are greater than or equal to m; step (b) is true due to the fact that n sampled queues are i.i.d. and thus the number of $\overline{Q}_l$'s that are greater than or equal to m follows binomial distribution with parameters n and $\overline{s}_m$; step (c) utilizes Binomial Theorem; step (d) is true by letting $d = j + l$; step (e) follows from the subset-of-a-subset identity [27] $\binom{n}{j}\binom{n-j}{d-j} = \binom{n}{d}\binom{d}{j}$.

By utilizing equation (40), we have

$$\begin{aligned}
&\sum_{i=1}^{k}\Pr\{\overline{Q}_{(i)}\geq m\}\\
=&\sum_{i=1}^{k}\sum_{d=n-i+1}^{n}\binom{n}{d}\overline{s}_m^d\sum_{j=n-i+1}^{d}\binom{d}{j}(-1)^{d-j}\\
\overset{(a)}{=}&\sum_{d=n-k+1}^{n}\binom{n}{d}\overline{s}_m^d\sum_{i=n+1-d}^{k}\sum_{j=n-i+1}^{d}\binom{d}{j}(-1)^{d-j}\\
\overset{(b)}{=}&\sum_{l=1}^{k}\binom{n}{n-k+l}\overline{s}_m^{n-k+l}\sum_{i=k+1-l}^{k}\sum_{j=n-i+1}^{n-k+l}\binom{n-k+l}{j}(-1)^{n-k+l-j},
\end{aligned}\tag{41}$$

where the step (a) is true by exchanging the order of the first and second summation; step (b) is true for $l=d-(n-k)$.

Next, we are going to show that

$$\sum_{i=k+1-l}^{k}\sum_{j=n-i+1}^{n-k+l}\binom{n-k+l}{j}(-1)^{n-k+l-j}=\binom{n-k+l-2}{l-1}(-1)^{l-1}.\tag{42}$$

Indeed,

$$\begin{aligned}
&\sum_{i=k+1-l}^{k}\sum_{j=n-i+1}^{n-k+l}\binom{n-k+l}{j}(-1)^{n-k+l-j}\\
\overset{(a)}{=}&\sum_{j=n-k+1}^{n-k+l}(k+j-n)\binom{n-k+l}{j}(-1)^{n-k+l-j}\\
\overset{(b)}{=}&\sum_{j'=1}^{l}j'\binom{n-k+l}{n-k+j'}(-1)^{l-j'}\\
=&\sum_{j=1}^{l}j\binom{n-k+l}{l-j}(-1)^{l-j}\\
=&l\binom{n-k+l}{0}+\sum_{j=1}^{l-1}j\binom{n-k+l}{l-j}(-1)^{l-j}\\
\overset{(c)}{=}&l\binom{n-k+l-1}{0}+\sum_{j=1}^{l-1}j\left(\binom{n-k+l-1}{l-j}+\binom{n-k+l-1}{l-j-1}\right)(-1)^{l-j}\\
=&\sum_{j=0}^{l-1}\binom{n-k+l-1}{j}(-1)^{j}\\
=&\binom{n-k+l-1}{0}+\sum_{j=1}^{l-1}\binom{n-k+l-1}{j}(-1)^{j}\\
\overset{(d)}{=}&\binom{n-k+l-2}{0}+\sum_{j=1}^{l-1}\left(\binom{n-k+l-2}{j}+\binom{n-k+l-2}{j-1}\right)(-1)^{j}\\
=&\binom{n-k+l-2}{l-1}(-1)^{l-1},
\end{aligned}\tag{43}$$

where the step (a) follows by switching the order of summations; step (b) is true by letting $j'=j-(n-k)$; steps (c) and (d) utilize the Pascal's rule, i.e.,

$$\binom{n}{k}=\binom{n-1}{k-1}+\binom{n-1}{k},$$

for all integers n,k: $1\leq k\leq n-1$.

By substituting (43) into (41), we have

$$\begin{aligned}&\sum_{i=1}^{k}\Pr\{\overline{Q}_{(i)}\geq m\}\\=&\sum_{l=1}^{k}\overline{s}_m^{n-k+l}\binom{n}{n-k+l}\binom{n-k+l-2}{l-1}(-1)^{l-1}\\=&f^{(n,k)}(\overline{s}_m),\end{aligned}\tag{44}$$

where $f^{(n,k)}(x)$ is defined in Lemma 1. By noting that $\Pr\{\overline{Q}_{(i)}=m\}=\Pr\{\overline{Q}_{(i)}\geq m\}-\Pr\{\overline{Q}_{(i)}\geq m-1\}$ and utilizing (44), we have the desired result.

APPENDIX C
PROOF OF LEMMA 2

By the definition of the function $f^{(n,k)}(x)$, we have

$$\begin{aligned}\left(f^{(n,k)}(x)\right)'=&\sum_{l=1}^{k}(-1)^{l-1}\binom{n}{n-k+l}\binom{n-k+l-2}{l-1}(n-k+l)x^{n-k+l-1}\\=&\sum_{l=1}^{k}(-1)^{l-1}\binom{n}{n-k+l}\binom{n-k+l}{n-k+l-1}\binom{n-k+l-2}{n-k-1}x^{n-k+l-1}\\\triangleq&\int_0^x g^{(n,k)}(z)dz,\end{aligned}\tag{45}$$

where $g^{(n,k)}(x)$ is defined as

$$g^{(n,k)}(x)\triangleq\sum_{l=1}^{k}(-1)^{l-1}\binom{n}{n-k+l}\binom{n-k+l}{n-k+l-1}\binom{n-k+l-1}{n-k+l-2}\binom{n-k+l-2}{n-k-1}x^{n-k+l-2}\ .$$

Once we obtain the closed-form expression for $g^{(n,k)}(x)$, we can easily get $f^{(n,k)}(x)$. Next, let's focus on $g^{(n,k)}(x)$.

$$\begin{aligned}g^{(n,k)}(x)\overset{(a)}{=}&C(n,k)\sum_{l=1}^{k}(-1)^{l-1}\binom{k-1}{l-1}x^{n-k+l-2}\\\overset{(b)}{=}&C(n,k)x^{n-k-1}\sum_{l'=0}^{k-1}(-1)^{l'}\binom{k-1}{l'}x^{l'}\\\overset{(c)}{=}&C(n,k)x^{n-k-1}(1-x)^{k-1},\end{aligned}\tag{46}$$

where the step (a) is true for $C(n,k)\triangleq\binom{n}{n-k-1}\binom{k+1}{1}\binom{k}{1}$ and utilizes the subset-of-a-subset identity shown as follows.

$$\begin{aligned}&\binom{n}{n-k+l}\binom{n-k+l}{n-k+l-1}\binom{n-k+l-1}{n-k+l-2}\binom{n-k+l-2}{n-k-1}\\=&\binom{n}{n-k-1}\binom{k+1}{1}\binom{k}{1}\binom{k-1}{l-1};\end{aligned}$$

step (b) is true by letting $l'=l-1$; step (c) utilizes Binomial Theorem.

By using (46), we have

$$\begin{aligned}\left(f^{(n,k)}(x)\right)'=&\int_0^x g^{(n,k)}(z)dz\\=&C(n,k)\int_0^x z^{n-k-1}(1-z)^{k-1}dz.\end{aligned}\tag{47}$$

Hence, $\left(f^{(n,k)}(x)\right)'>0$ for any $x\in(0,1]$ and therefore $f^{(n,k)}(x)$ is strictly increasing in $x\in[0,1]$.

Moreover,

$$\left(f^{(n,k)}(x)\right)''=C(n,k)x^{n-k-1}(1-x)^{k-1}\geq 0\tag{48}$$

holds for any $x\in[0,1]$, which implies that $f^{(n,k)}(x)$ is also convex on the interval $[0,1]$.

In addition, it is easy to see that $f^{(n,k)}(0)=0$ from the definition of $f^{(n,k)}(x)$, and

$$\begin{aligned} f^{(n,k)}(1) =& \int_0^1 \left(f^{(n,k)}(x)\right)' dx \\ =& C(n,k)\int_0^1 dx \int_0^x z^{n-k-1}(1-z)^{k-1}dz \\ \overset{(a)}{=}& C(n,k)\int_0^1 z^{n-k-1}(1-z)^{k}dz \\ \overset{(b)}{=}& k, \end{aligned} \tag{49}$$

where the step (a) interchanges the order of integrals; step (b) uses the fact that

$$\int_0^1 z^a(1-z)^b dx = \frac{a!b!}{(a+b+1)!}, \tag{50}$$

for any non-negative integers a and b, and the definition of $C(n,k)$.

Furthermore, since $\left(f^{(n,k)}(x)\right)'' \geq 0$ for any $x\in[0,1]$, $\left(f^{(n,k)}(x)\right)'$ is non-decreasing on the interval $[0,1]$, which implies

$$\left(f^{(n,k)}(x)\right)' \leq \left(f^{(n,k)}(1)\right)' = C(n,k)\int_0^1 z^{n-k-1}(1-z)^{k-1}dz = n, \tag{51}$$

where the last step again uses (50) and the definition of $C(n,k)$.

APPENDIX D
PROOF OF LEMMA 3

We use mathematical induction to show this lemma. First, note that $\overline{s}_0=\hat{s}_0=1$. Assume that $\overline{s}_m\leq\hat{s}_m$ for some $m\geq 0$. Then, we have

$$\overline{s}_{m+1} = \frac{\lambda}{k} f^{(nk,k)}(\overline{s}_m) \overset{(a)}{\leq} \lambda \overline{s}_m^n \overset{(b)}{\leq} \lambda \hat{s}_m^n = \hat{s}_{m+1}, \tag{52}$$

where the step (a) uses Lemma 6; step (b) follows from the induction hypothesis.

Lemma 6:

$$\frac{1}{k} f^{(n,k)}(x) \leq x^{n/k} \tag{53}$$

holds for any $x\in[0,1]$.

Proof: Since $f^{(n,k)}(0)=0$ (cf. Lemma 2), inequality (53) holds when $x=0$. In the rest of the proof, we consider the case when $x\in(0,1]$. We will show that $f^{(n,k)}(x)/k \leq x^{n/k}$, which is equivalent to proving

$$h(x) \triangleq \frac{1}{k}\frac{f^{(n,k)}(x)}{x^{n/k}} \leq 1. \tag{54}$$

Since $h(1)=f^{(n,k)}(1)/k=1$ (cf. Lemma 2), it is sufficient to show that $h'(x)\geq 0$ for any $x\in(0,1]$. Noting that

$$h'(x) = \frac{1}{k}\frac{x(f^{(n,k)}(x))' - \frac{n}{k}f^{(n,k)}(x)}{x^{n/k+1}}, \tag{55}$$

It is equivalent to showing that

$$x(f^{(n,k)}(x))' \geq \frac{n}{k} f^{(n,k)}(x), \tag{56}$$

holds for any $x\in(0,1]$.

According to (47) in the proof of Lemma 2, we have

$$\begin{aligned} & (f^{(n,k)}(x))' \\ =& C(n,k)\int_0^x z^{n-k-1}(1-z)^{k-1}dz \\ =& C(n,k)\sum_{l=0}^{k-1}\binom{k-1}{l}(-1)^l \int_0^x z^{n-k-1+l}dz \\ =& C(n,k)\sum_{l=0}^{k-1}\binom{k-1}{l}(-1)^l \frac{x^{n-k+l}}{n-k+l}, \end{aligned} \tag{57}$$

where we recall that $C(n,k) \triangleq \binom{n}{k+1}\binom{k+1}{1}\binom{k}{1}$, and the second step follows from the Binomial Theorem. Hence, we have

$$\begin{aligned} & f^{(n,k)}(x) \\ =& \int_0^x (f^{(n,k)}(z))' dz \\ =& C(n,k) \sum_{l=0}^{k-1} \binom{k-1}{l} (-1)^l \frac{1}{n-k+l} \frac{x^{n-k+l+1}}{n-k+l+1}. \end{aligned} \tag{58}$$

By substituting (57) and (58) into (56), it is equivalent to showing that

$$w(x) \triangleq \sum_{l=0}^{k-1} \binom{k-1}{l} (-1)^l \frac{1}{n-k+l} \frac{(n/k-1)(k-1)+l}{n-k+l+1} x^{n-k+l+1} \geq 0.$$

Next, we will study some basic properties of $w(x)$. First, we note that

$$\begin{aligned} & w''(x) \\ =& x^{n-k-1} \sum_{l=0}^{k-1} \binom{k-1}{l} (-1)^l \left((n/k-1)(k-1)+l\right) x^l \\ =& x^{n-k-1} \left((n/k-1)(k-1)(1-x)^{k-1} - x(k-1)(1-x)^{k-2}\right) \\ =& (k-1) x^{n-k-1} (1-x)^{k-2} \left((n/k-1) - nx/k\right), \end{aligned} \tag{59}$$

where the second last step follows from the Binomial Theorem and the fact that

$$\sum_{l=0}^{k-1} \binom{k-1}{l} l(-x)^l = -x(k-1)(1-x)^{k-2}.$$

Note that $w''(x) \geq 0$ if $x \in (0, 1-k/n]$, and $w''(x) \leq 0$ if $x \in (1-k/n, 1]$, which implies that $w'(x)$ is non-decreasing on the interval $(0, 1-k/n]$, and non-increasing on $(1-k/n, 1]$.

Since

$$w'(x) = \int_0^x w''(z) dz = (k-1) \int_0^x z^{n-k-1} (1-z)^{k-2} \left(\frac{n}{k} - 1 - \frac{n}{k} z\right) dz, \tag{60}$$

we have $w'(0) = 0$ and

$$\begin{aligned} w'(1) =& (k-1) \left(\left(\frac{n}{k} - 1\right) \int_0^1 z^{n-k-1} (1-z)^{k-2} dz - \frac{n}{k} \int_0^1 z^{n-k} (1-z)^{k-2} dz \right) \\ =& \frac{(k-1)!(n-k)!}{(n-2)!k} \left(1 - \frac{n}{n-1}\right) < 0, \end{aligned} \tag{61}$$

where the second step uses (50). Therefore, there must exist a point $x_0 \in (1-k/n, 1)$ such that $w'(x) \geq 0$ for any $x \in (0, x_0]$ and $w'(x) < 0$ for any $x \in (x_0, 1]$, which implies that $w(x)$ is non-decreasing on the interval $(0, x_0]$ and non-increasing on $(x_0, 1]$.

Since $w(0) = 0$, we have $w(x) \geq 0$ for any $x \in (0, 1]$ if $w(1) \geq 0$. Indeed, we have

$$\begin{aligned} w(1) =& \int_0^1 w'(x) dx \\ =& (k-1) \int_0^1 dx \int_0^x z^{n-k-1} (1-z)^{k-2} \left(\left(\frac{n}{k} - 1\right) - \frac{n}{k} z \right) dz \\ \overset{(a)}{=}& (k-1) \left(\left(\frac{n}{k} - 1\right) \int_0^1 z^{n-k-1} (1-z)^{k-1} dz - \frac{n}{k} \int_0^1 z^{n-k} (1-z)^{k-1} dz \right) \\ \overset{(b)}{=}& 0, \end{aligned} \tag{62}$$

where the step (a) interchanges the order of integrals; step (b) uses (50). ■

APPENDIX E
PROOF OF LEMMA 4

First, recall that $\overline{s}_m$ and $\hat{s}_m$ are the probabilities that steady-state queue length is at least m under (nk,k) and $(n,1)$ codes in the large-system limit, respectively. We note that

$$\begin{aligned}
&\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}\left[\overline{Q}_{(i)}^{(nk,k)}\right]\\
\overset{(a)}{=}&\frac{1}{k}\sum_{i=1}^{k}\sum_{m=1}^{\infty}\Pr\left\{\overline{Q}_{(i)}^{(nk,k)}\geq m\right\}\\
\overset{(b)}{=}&\frac{1}{k}\sum_{m=1}^{\infty}f^{(nk,k)}(\overline{s}_m),
\end{aligned}\tag{63}$$

where step (a) uses the fact that $\mathbb{E}[Z]=\sum_{m=1}^{\infty}\Pr\{Z\geq m\}$ for any non-negative integer-valued random variable Z; step (b) interchanges the order of summations (since $\Pr\left\{\overline{Q}_{(i)}^{(nk,k)}\geq m\right\}\geq 0,\forall i,m$) and follows from Lemma 1. By Proposition 3, under (nk,k) code,

$$\overline{s}_{m+1}=\frac{\lambda}{k}f^{(nk,k)}(\overline{s}_m),\forall m=0,1,2,\cdots\tag{64}$$

By combining (63) and (64), we have

$$\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}\left[\overline{Q}_{(i)}^{(nk,k)}\right]=\frac{1}{\lambda}\sum_{m=1}^{\infty}\overline{s}_{m+1}\tag{65}$$

On the other hand, under $(n,1)$ code,

$$\begin{aligned}
\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]=&\sum_{m=1}^{\infty}\Pr\left\{\overline{Q}_{(1)}^{(n,1)}\geq m\right\}\\
=&\sum_{m=1}^{\infty}\hat{s}_m^n\\
=&\frac{1}{\lambda}\sum_{m=1}^{\infty}\hat{s}_{m+1},
\end{aligned}\tag{66}$$

where the last step uses the fact that $\hat{s}_{m+1}=\lambda\hat{s}_m^n,\forall m=0,1,2,\cdots$ under $(n,1)$ code according to Proposition 3. Hence, the desired result follows from (65), (66), and Lemma 3.

APPENDIX F
PROOF OF LEMMA 5

We first note that $\overline{s}_m$ and $\hat{s}_m$ are the probabilities that steady-state queue length is at least m under (nk,k) and $(n,1)$ codes in the large-system limit, respectively. Therefore, according to Proposition 3, we have

$$\hat{s}_m=\lambda^{\frac{n^m-1}{n-1}},\forall m=0,1,2,\cdots\tag{67}$$

According to equation (66), we have

$$\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]=\sum_{m=1}^{\infty}\hat{s}_m^n=\sum_{m=1}^{\infty}\lambda^{\frac{n^m-1}{n-1}n},\tag{68}$$

which implies that

$$\lim_{\lambda\uparrow 1}\frac{\mathbb{E}\left[\overline{Q}_{(1)}^{(n,1)}\right]}{-\log(1-\lambda)}=\lim_{\lambda\uparrow 1}\frac{\sum_{m=1}^{\infty}\lambda^{\frac{n^m-1}{n-1}n}}{-\log(1-\lambda)}=\frac{1}{\log n},\tag{69}$$

where the last step utilizes Lemma 7.

Lemma 7:

$$\lim_{\lambda\uparrow 1}\frac{\sum_{m=1}^{\infty}\lambda^{\frac{n^m-1}{n-1}a}}{-\log(1-\lambda)}=\frac{1}{\log n},\tag{70}$$

holds for any real number $a>0$.

The proof of Lemma 7 is similar to [16, Theorem 3.9] and is provided next for completeness.

Proof:

$$\begin{aligned}
&\lim_{\lambda\uparrow 1}\frac{\sum_{m=1}^{\infty}\lambda^{\frac{n^m-1}{n-1}a}}{-\log(1-\lambda)}\\
=&\lim_{\lambda\uparrow 1}\frac{\sum_{m=1}^{\infty}\left(\lambda^{\frac{a}{n-1}}\right)^{n^m}}{-\log(1-\lambda)}\frac{1}{\lambda^{\frac{a}{n-1}}}\\
\overset{(a)}{=}&\lim_{\lambda'\uparrow 1}\frac{\sum_{m=1}^{\infty}(\lambda')^{n^m}}{-\log(1-\lambda')}\frac{\log(1-\lambda')}{\log(1-\lambda)}\frac{1}{\lambda^{\frac{a}{n-1}}}\\
\overset{(b)}{=}&\lim_{\lambda\uparrow 1}\frac{\sum_{m=1}^{\infty}\lambda^{n^m}}{-\log(1-\lambda)}\\
\overset{(c)}{=}&\frac{1}{\log n},
\end{aligned}\tag{71}$$

where the step (a) is true by setting $\lambda'=\lambda^{\frac{a}{n-1}}$; step (b) follows from the fact that the last two terms go to 1 as $\lambda\uparrow 1$; step (c) utilizes [16, Lemma 3.10]. ■

Next, we consider the heavy-traffic behavior of the expression $\sum_{i=1}^{k}\frac{1}{k-i+1}\mathbb{E}\left[\overline{Q}_{(i)}^{(nk,k)}\right]$. First, we note that

$$\begin{aligned}
&\sum_{i=1}^{k}\frac{1}{k-i+1}\mathbb{E}\left[\overline{Q}_{(i)}^{(nk,k)}\right]\\
\overset{(a)}{=}&\sum_{i=1}^{k}\frac{1}{k-i+1}\sum_{m=1}^{\infty}\Pr\left\{\overline{Q}_{(i)}^{(nk,k)}\geq m\right\}\\
\overset{(b)}{=}&\sum_{m=1}^{\infty}\sum_{i=1}^{k}\frac{1}{k-i+1}\Pr\left\{\overline{Q}_{(i)}^{(nk,k)}\geq m\right\},
\end{aligned}\tag{72}$$

where the step (a) uses the fact that $\mathbb{E}[Z]=\sum_{m=1}^{\infty}\Pr\{Z\geq m\}$ for any non-negative integer-valued random variable Z; step (b) interchanges the order of summations since $\Pr\left\{\overline{Q}_{(i)}^{(nk,k)}\geq m\right\}\geq 0,\forall i,m$.

Next, we express $\sum_{i=1}^{k}\frac{1}{k-i+1}\Pr\left\{\overline{Q}_{(i)}^{(nk,k)}\geq m\right\}$ as a function of the queue length distribution under the (nk,k) code, where we follows a similar procedure as in the proof of Lemma 1.

$$\begin{aligned}
&\sum_{i=1}^{k}\frac{1}{k-i+1}\Pr\left\{\overline{Q}_{(i)}^{(nk,k)}\geq m\right\}\\
\overset{(a)}{=}&\sum_{i=1}^{k}\frac{1}{k-i+1}\sum_{d=nk-i+1}^{nk}\binom{nk}{d}\overline{s}_m^d\sum_{j=nk-i+1}^{d}\binom{d}{j}(-1)^{d-j}\\
\overset{(b)}{=}&\sum_{d=nk-k+1}^{nk}\binom{nk}{d}\overline{s}_m^d\sum_{i=nk+1-d}^{k}\sum_{j=nk-i+1}^{d}\frac{1}{k-i+1}\binom{d}{j}(-1)^{d-j}\\
\overset{(c)}{=}&\sum_{l=1}^{k}\binom{nk}{nk-k+l}\overline{s}_m^{nk-k+l}\sum_{i=k+1-l}^{k}\sum_{j=nk-i+1}^{nk-k+l}\frac{1}{k-i+1}\binom{nk-k+l}{j}(-1)^{nk-k+l-j},
\end{aligned}\tag{73}$$

where the step (a) follows from equation (40); step (b) interchanges the order of the first and second summation; step (c) is true for $l=d-(nk-k)$.

Next, we are going to simplify the term $\sum_{i=k+1-l}^{k}\sum_{j=nk-i+1}^{nk-k+l}\frac{1}{k-i+1}\binom{nk-k+l}{j}(-1)^{nk-k+l-j}$.

$$\begin{aligned}
&\sum_{i=k+1-l}^{k}\sum_{j=nk-i+1}^{nk-k+l}\frac{1}{k-i+1}\binom{nk-k+l}{j}(-1)^{nk-k+l-j}\\
\overset{(a)}{=}&\sum_{j=nk-k+1}^{nk-k+l}\binom{nk-k+l}{j}(-1)^{nk-k+l-j}H(j-(nk-k))\\
\overset{(b)}{=}&\sum_{j'=1}^{l}\binom{nk-k+l}{nk-k+j'}(-1)^{l-j'}H(j')\\
=&\sum_{j=1}^{l}\binom{nk-k+l}{l-j}(-1)^{l-j}H(j)\\
=&H(l)\binom{nk-k+l}{0}+\sum_{j=1}^{l-1}H(j)\binom{nk-k+l}{l-j}(-1)^{l-j}\\
\overset{(c)}{=}&H(l)\binom{nk-k+l-1}{0}+\sum_{j=1}^{l-1}H(j)\left(\binom{nk-k+l-1}{l-j}+\binom{nk-k+l-1}{l-j-1}\right)(-1)^{l-j}\\
=&\sum_{j=1}^{l}\frac{1}{j}\binom{nk-k+l-1}{l-j}(-1)^{l-j},
\end{aligned}\tag{74}$$

where step (a) is true by switching the order of summations and recalling that $H(m)\triangleq\sum_{l=1}^{m}1/l$ is the l^{th} harmonic number; step (b) is true for letting $j'=j-(nk-k)$; step (c) utilizes the Pascal's rule, i.e., $\binom{n}{k}=\binom{n-1}{k-1}+\binom{n-1}{k}$ for all integers n and k satisfying $1\leq k\leq n-1$.

By substituting (74) into (73), we have

$$\sum_{i=1}^{k}\frac{1}{k-i+1}\Pr\left\{\overline{Q}_{(l)}^{(nk,k)}\geq m\right\}=\varphi(\overline{s}_m),\tag{75}$$

where $\varphi(x)=\sum_{l=1}^{k}C_l x^{nk-k+l}, x\in[0,1]$ and $C_l\triangleq\binom{nk}{nk-k+l}\sum_{j=1}^{l}\frac{1}{j}\binom{nk-k+l-1}{l-j}(-1)^{l-j}$.

The next lemma reveals two important properties of the function $\varphi(x)$, which is important in establishing the desired result.

Lemma 8: The function $\varphi(x)$ has the following two properties:

(i) The function $\varphi(x)$ is increasing on the interval $[0,1]$;

(ii) $\varphi(0)=0$ and $\varphi(1)=H(k)$.

The proof of Lemma 8 is available in Appendix G.

Since $0\leq\overline{s}_m\leq\hat{s}_m\leq 1,\forall m\geq 0$ (cf. Lemma 3), according to Lemma 8, we have $\varphi(\overline{s}_m)\leq\varphi(\hat{s}_m)$ and thus

$$\sum_{i=1}^{k}\frac{1}{k-i+1}\Pr\left\{\overline{Q}_{(l)}^{(nk,k)}\geq m\right\}\leq\varphi(\hat{s}_m).\tag{76}$$

This combined with (72) implies that

$$\begin{aligned}
&\sum_{i=1}^{k}\frac{1}{k-i+1}\mathbb{E}\left[\overline{Q}_{(i)}^{(nk,k)}\right]\\
\leq&\sum_{m=1}^{\infty}\varphi(\hat{s}_m)\\
=&\sum_{l=1}^{k}C_l\sum_{m=1}^{\infty}\hat{s}_m^{nk-k+l}\\
=&\sum_{l=1}^{k}C_l\sum_{m=1}^{\infty}\lambda^{\frac{n^m-1}{n-1}(nk-k+l)},
\end{aligned}\tag{77}$$

where we use (67), i.e., $\hat{s}_m=\lambda^{\frac{n^m-1}{n-1}},\forall m=0,1,2,\cdots$.

Hence, we have

$$\begin{aligned}
&\lim_{\lambda\uparrow 1}\frac{\sum_{l=1}^{k}\frac{1}{k-l+1}\mathbb{E}\left[\overline{Q}_{(l)}^{(nk,k)}\right]}{-\log(1-\lambda)}\\
\leq&\sum_{l=1}^{k}C_l\lim_{\lambda\uparrow 1}\frac{\sum_{m=1}^{\infty}\lambda^{\frac{n^m-1}{n-1}(nk-k+l)}}{-\log(1-\lambda)}\\
\overset{(a)}{=}&\frac{1}{\log n}\sum_{l=1}^{k}C_l\\
=&\frac{1}{\log n}\varphi(1)\\
\overset{(b)}{=}&\frac{H(k)}{\log n},
\end{aligned}\tag{78}$$

where the step (a) follows from Lemma 7 ; step (b) follows from Lemma 8. Combining (78) and (69), we have the desired result.

Appendix G
Proof of Lemma 8

By the definition of the function $\varphi(x)$, we have

$$\begin{aligned}
\varphi'(x)=&\sum_{l=1}^{k}\sum_{j=1}^{l}x^{nk-k+l-1}(nk-k+l)\binom{nk}{nk-k+l}\binom{nk-k+l-1}{l-j}\frac{(-1)^{l-j}}{j}\\
\overset{(a)}{=}&\sum_{j=1}^{k}\sum_{l=j}^{k}x^{nk-k+l-1}\binom{nk}{nk-k+l}\binom{nk-k+l}{nk-k+l-1}\binom{nk-k+l-1}{nk-k+j-1}\frac{(-1)^{l-j}}{j}\\
\overset{(b)}{=}&\sum_{j=1}^{k}\frac{1}{j}\binom{nk}{nk-k+j-1}\binom{k+1-j}{1}\sum_{l=j}^{k}\binom{k-j}{l-j}(-1)^{l-j}x^{nk-k+l-1},
\end{aligned}\tag{79}$$

where the step (a) switches the order of summations; step (b) utilizes the subset-of-a-subset identity stated below.

$$\binom{nk}{nk-k+l}\binom{nk-k+l}{nk-k+l-1}\binom{nk-k+l-1}{nk-k+j-1}=\binom{nk}{nk-k+j-1}\binom{k+1-j}{1}\binom{k-j}{l-j}.$$

Since

$$\begin{aligned}
&\sum_{l=j}^{k}\binom{k-j}{l-j}(-1)^{l-j}x^{nk-k+l-1}\\
=&\sum_{l'=0}^{k-j}\binom{k-j}{l'}(-1)^{l'}x^{l'}x^{nk-k-1+j}\\
=&x^{nk-k-1+j}(1-x)^{k-j},
\end{aligned}\tag{80}$$

we have

$$\varphi'(x)=\sum_{j=1}^{k}\frac{1}{j}\binom{nk}{nk-k+j-1}\binom{k+1-j}{1}x^{nk-k-1+j}(1-x)^{k-j}.\tag{81}$$

Since $\varphi'(x)\geq 0$ for all $x\in[0,1]$, $\varphi(x)$ is increasing on the interval $[0,1]$.

In addition, we have

$$\begin{aligned}
\varphi(x)=&\int_0^x\varphi'(z)dz\\
=&\sum_{j=1}^{k}\frac{1}{j}\binom{nk}{nk-k+j-1}\binom{k+1-j}{1}\int_0^x z^{nk-k-1+j}(1-z)^{k-j}dz.
\end{aligned}\tag{82}$$

Therefore, $\varphi(0)=0$ and

$$\begin{aligned}\varphi(1)=&\sum_{j=1}^{k}\frac{1}{j}\binom{nk}{nk-k+j-1}\binom{k+1-j}{1}\int_0^1 z^{nk-k-1+j}(1-z)^{k-j}dz\\ =&\sum_{j=1}^{k}\frac{1}{j}\binom{nk}{nk-k+j-1}\binom{k+1-j}{1}\frac{(nk-k-1+j)!(k-j)!}{(nk)!}\\ =&H(k),\end{aligned}\tag{83}$$

where the second step utilizes (50).

APPENDIX H
PROOF OF PROPOSITION 4

In the rest of the proof, we omit the superscript (n,k) associated with $f^{(n,k)}(x)$, $F_m^{(n,k)}(\mathbf{x})$ and $\mathbf{F}^{(n,k)}(\mathbf{x})$ for simplicity. We first show that $\mathbf{F}(\mathbf{x})$ is Lipschitz, which implies that the solution of the differential equations (27) is continuous.

Lemma 9: The drift function $\mathbf{F}(\mathbf{s})$ is Lipschitz, i.e., there exists a constant $G>0$ such that for any $\mathbf{x},\mathbf{y}\in\mathcal{S}$,

$$\|\mathbf{F}(\mathbf{x})-\mathbf{F}(\mathbf{y})\|_1\le G\|\mathbf{x}-\mathbf{y}\|_1,\tag{84}$$

where $\|\mathbf{z}\|_1\triangleq\sum_{m=0}^{\infty}|z_m|$ is the l_1 norm of the vector $\mathbf{z}=(z_m)_{m\ge0}\in\mathcal{S}$.

Proof:

$$\begin{aligned}&\|\mathbf{F}(\mathbf{x})-\mathbf{F}(\mathbf{y})\|_1\\ =&\sum_{m=1}^{\infty}|F_m(\mathbf{x})-F_m(\mathbf{y})|\\ =&\sum_{m=1}^{\infty}\left|\lambda(f(x_{m-1})-f(x_m))-k(x_m-x_{m+1})-\lambda(f(y_{m-1})-f(y_m))+k(y_m-y_{m+1})\right|\\ \le&2k\sum_{m=1}^{\infty}|x_m-y_m|+2\lambda\sum_{m=1}^{\infty}|f(x_m)-f(y_m)|\\ \le&G\|\mathbf{x}-\mathbf{y}\|_1,\end{aligned}\tag{85}$$

where the last step utilizes the fact that the function $f(x)$ has the bounded derivative (cf. Lemma 2) and is true for $G\triangleq 2(k+n\lambda)$. ∎

The rest of the proof basically follows from [15, Section 3] and utilizes the monotone property of the function $f(x)$ (cf. Lemma 2). In order to study the differential equations (25), we first consider the following truncated system of differential equations:

$$\frac{ds_m(t)}{dt}=\lambda\left(f(s_{m-1}(t))-f(s_m(t))\right)-k\left(s_m(t)-s_{m+1}(t)\right),\forall m=1,2,\cdots,M,\tag{86}$$

where the solution $\mathbf{s}(t)=\{s_m(t)\}_{m=1}^{M}$ satisfies: (i) $s_0(t)\equiv1$ and $s_{M+1}(t)\equiv c\ge0$ for any $t\ge0$; (ii) $s_m(0)=\eta_m,\forall m=1,2,\cdots,M$.

Lemma 10: If the initial values of the truncated system satisfies

$$1=s_0(t)\ge s_1(t)\ge\cdots\ge s_M(t)\ge s_{M+1}(t),\tag{87}$$

then the solution of this system also satisfies this condition for all $t\ge0$.

Proof: Due to the continuous dependence of a solution on the initial values, it is sufficient to consider initial values where strict inequalities hold, i.e., $1=s_0(t)>s_1(t)>\cdots>s_M(t)>s_{M+1}(t)$. Assume that strict inequalities hold for all $t<t_1$, and are broken at $t=t_1$. Let $\widetilde{m}$ be the largest index with $s_{\widetilde{m}}(t_1)=s_{\widetilde{m}+1}(t_1)$, i.e., $s_{\widetilde{m}-1}(t_1)>s_{\widetilde{m}}(t_1)=s_{\widetilde{m}+1}(t_1)>s_{\widetilde{m}+2}(t_1)$. Therefore, we have

$$\begin{aligned}\frac{ds_{\widetilde{m}}(t_1)}{dt}=&\lambda\left(f(s_{\widetilde{m}-1}(t_1))-f(s_{\widetilde{m}}(t_1))\right)-k\left(s_{\widetilde{m}}(t_1)-s_{\widetilde{m}+1}(t_1)\right)\\ =&\lambda\left(f(s_{\widetilde{m}-1}(t_1))-f(s_{\widetilde{m}}(t_1))\right)>0,\end{aligned}\tag{88}$$

where the last step uses the strict increasing property of $f(x)$ (cf. Lemma 2). On the other hand, we have

$$\begin{aligned}\frac{ds_{\widetilde{m}+1}(t_1)}{dt}=&\lambda\left(f(s_{\widetilde{m}}(t_1))-f(s_{\widetilde{m}+1}(t_1))\right)-k\left(s_{\widetilde{m}+1}(t_1)-s_{\widetilde{m}+2}(t_1)\right)\\ =&-k\left(s_{\widetilde{m}+1}(t_1)-s_{\widetilde{m}+2}(t_1)\right)<0.\end{aligned}\tag{89}$$

Therefore, we have $\frac{ds_{\widetilde{m}}(t_1)}{dt} > \frac{ds_{\widetilde{m}+1}(t_1)}{dt}$. Since $\mathbf{s}(t)$ is continuous with respect to t, there exists a $t_0 < t_1$ such that

$$\frac{ds_{\widetilde{m}}(t)}{dt} > \frac{ds_{\widetilde{m}+1}(t)}{dt}, \forall t \in (t_0, t_1). \tag{90}$$

By the assumption, we also have $s_{\widetilde{m}}(t) > s_{\widetilde{m}+1}(t), \forall t \in (t_0, t_1)$. Therefore, we have

$$s_{\widetilde{m}}(t_1) - s_{\widetilde{m}+1}(t_1) = s_{\widetilde{m}}(t_0) - s_{\widetilde{m}+1}(t_0) + \int_{t_0}^{t_1} \left(\frac{ds_{\widetilde{m}}(t)}{dt} - \frac{ds_{\widetilde{m}+1}(t)}{dt} \right) dt > 0,$$

which implies $s_{\widetilde{m}}(t_1) > s_{\widetilde{m}+1}(t_1)$. This contradicts with the assumption that $s_{\widetilde{m}}(t_1) = s_{\widetilde{m}+1}(t_1)$. ■

Lemma 11: Let $\mathbf{s}^{(1)}(t)$ and $\mathbf{s}^{(2)}(t)$ be the two solutions of the truncated system. If $s_m^{(1)}(0) \geq s_m^{(2)}(0), \forall m = 1, 2, \cdots, M$ and $s_{M+1}^{(1)}(t) \geq s_{M+1}^{(2)}(t)$, $t \geq 0$. Then, $s_m^{(1)}(t) \geq s_m^{(2)}(t), \forall m = 1, 2, \cdots, M$, $\forall t \geq 0$.

Proof: It is again sufficient to consider the case where strict inequalities hold, i.e., $s_m^{(1)}(0) > s_m^{(2)}(0)$, $\forall m = 1, 2, \cdots, M$ and $s_{M+1}^{(1)}(t) > s_{M+1}^{(2)}(t), \forall t \geq 0$, and show that $s_m^{(1)}(t) > s_m^{(2)}(t), m = 1, 2, \cdots, M, t > 0$. Assume that strict inequalities hold for all $t < t_1$, and are broken at $t = t_1$. Let $\widetilde{m}$ be the largest index with $s_{\widetilde{m}}^{(1)}(t_1) = s_{\widetilde{m}}^{(2)}(t_1)$, which implies that $s_{\widetilde{m}-1}^{(1)}(t_1) \geq s_{\widetilde{m}-1}^{(2)}(t_1)$ and $s_{\widetilde{m}+1}^{(1)}(t_1) > s_{\widetilde{m}+1}^{(2)}(t_1)$. Therefore, we have

$$\begin{aligned} &\frac{ds_{\widetilde{m}}^{(1)}(t_1)}{dt} - \frac{ds_{\widetilde{m}}^{(2)}(t_1)}{dt} \\ =&\lambda \left(f(s_{\widetilde{m}-1}^{(1)}(t_1)) - f(s_{\widetilde{m}-1}^{(2)}(t_1)) \right) + k \left(s_{\widetilde{m}+1}^{(1)}(t_1) - s_{\widetilde{m}+1}^{(2)}(t_1) \right) > 0, \end{aligned} \tag{91}$$

where the last step again uses the monotone property of $f(x)$ (cf. Lemma 2). Since $\mathbf{s}(t)$ is continuous with respect to t, there exists a $t_0 < t_1$ such that

$$\frac{ds_{\widetilde{m}}^{(1)}(t)}{dt} - \frac{ds_{\widetilde{m}}^{(2)}(t)}{dt} > 0, \forall t \in (t_0, t_1). \tag{92}$$

By the assumption, we also have $s_{\widetilde{m}}^{(1)}(t) > s_{\widetilde{m}}^{(2)}(t), \forall t \in (t_0, t_1)$. Therefore, we have

$$\begin{aligned} &s_{\widetilde{m}}^{(1)}(t_1) - s_{\widetilde{m}}^{(2)}(t_1) \\ =&s_{\widetilde{m}}^{(1)}(t_0) - s_{\widetilde{m}}^{(2)}(t_0) + \int_{t_0}^{t_1} \left(\frac{ds_{\widetilde{m}}^{(1)}(t)}{dt} - \frac{ds_{\widetilde{m}}^{(2)}(t)}{dt} \right) dt > 0, \end{aligned}$$

which implies that $s_{\widetilde{m}}^{(1)}(t_1) > s_{\widetilde{m}}^{(2)}(t_1)$. This contradicts with the assumption that $s_{\widetilde{m}}^{(1)}(t_1) = s_{\widetilde{m}}^{(2)}(t_1)$. ■

We are ready to consider the original system of differential equations.

Lemma 12: If the initial point $\boldsymbol{\eta} \in \mathcal{S}$, then there exists a unique solution of the original system of differential equations (25). This solution is the limit of a sequence of solutions of the truncated system with $s_{M+1}(t) = 0$ as $M \to \infty$.

Proof: Let $\mathbf{s}^{(M)}(t), M = 1, 2, \cdots$ be the solution of the truncated system with $s_{M+1}^{(M)} = 0$. By Lemma 10, $s_m^{(M)}$ satisfies (87) and therefore $s_{M+1}^{(M+1)}(t) \geq s_{M+1}^{(M)}(t) = 0$. By Lemma 11, for any fixed t and $m < M$, $s_m^{(M)}(t)$ is non-decreasing as M increases. Therefore, $\lim_{M\to\infty} s_m^{(M)}(t) = s_m(t)$ exists and $\mathbf{s}(t) = \{s_m(t)\}_{m=0}^{\infty} \in \mathcal{S}$, since $\mathcal{S}$ is compact. Turning from differential equations to integral ones, we can confirm that $s_m(t)$ is the solution of the original system. The uniqueness of this solution can be proved by the Picard successive-approximation method. ■

Lemma 13: Let $\mathbf{s}^{(1)}(t)$ and $\mathbf{s}^{(2)}(t)$ be the two solutions of the original system. If $s_m^{(1)}(0) \geq s_m^{(2)}(0), \forall m = 1, 2, \cdots$. Then, $s_m^{(1)}(t) \geq s_m^{(2)}(t), \forall m = 1, 2, \cdots, \forall t \geq 0$.

Proof: The proof directly follows from Lemmas 11 and 12. ■

Lemma 14: If the initial condition $\boldsymbol{\eta} \in \mathcal{S}$ satisfies either $\eta_m \leq \overline{s}_m, \forall m \geq 0$ or $\eta_m \geq \overline{s}_m, \forall m \geq 0$, then $\mathbf{s}(t)$ converges to the unique equilibrium point $\overline{\mathbf{s}}$.

Proof: Recall that $\nu_m(\mathbf{s}) \triangleq \sum_{j=m}^{\infty} s_j$. Then, we have

$$\begin{aligned} \frac{d\nu_m(\mathbf{s}(t))}{dt} =&\lambda f(s_{m-1}(t)) - k s_m(t) \\ =&\lambda \left(f(s_{m-1}(t)) - f(\overline{s}_{m-1}) \right) - k(s_m(t) - \overline{s}_m). \end{aligned} \tag{93}$$

Therefore, we have

$$\nu_m(\mathbf{s}(t)) - \nu_m(\boldsymbol{\eta}) = \int_0^t \left(\lambda \left(f(s_{m-1}(\tau)) - f(\overline{s}_{m-1}) \right) - k(s_m(\tau) - \overline{s}_m) \right) d\tau. \tag{94}$$

(i) The initial condition $\boldsymbol{\eta} \in \mathcal{S}$ satisfies $\eta_m \leq \overline{s}_m, \forall m \geq 0$:

According to Lemma 13, we have $s_m(t) \leq \overline{s}_m, \forall m, \forall t \geq 0$. Hence, $\nu_m(\mathbf{s}(t)) \leq \nu_1(\mathbf{s}(t)) \leq \sum_{l=1}^{\infty} \overline{s}_l < \infty$, $\forall m \geq 1$, which implies that $\nu_m(\mathbf{s}(t))$ is uniformly bounded with respect to $t \geq 0$.

Note that we have the desired result if we can show that $\int_0^{\infty}(s_m(t) - \overline{s}_m)dt > -\infty$. Next, we use mathematical induction to show that

$$\int_0^{\infty}(s_m(t) - \overline{s}_m)dt > -\infty \tag{95}$$

holds for any non-negative integer m.

It is obvious that (95) holds for $m = 0$. Assume that (95) holds for $m-1$. Then, we have

$$\begin{aligned}\int_0^{\infty}\left(f(s_{m-1}(t)) - f(\overline{s}_{m-1})\right)dt &\overset{(a)}{=} \int_0^{\infty} f'(z)(s_{m-1}(t) - \overline{s}_{m-1})dt \\ &\overset{(b)}{\geq} n\int_0^{\infty}(s_{m-1}(t) - \overline{s}_{m-1})dt \\ &> -\infty,\end{aligned} \tag{96}$$

where the step (a) uses the Mean-Value Theorem for some z between $s_{m-1}(t)$ and $\overline{s}_{m-1}$; step (b) utilizes the boundedness of the derivative of the function $f(x)$ (cf. Lemma 2). Since $\nu_m(\mathbf{s}(t))$ is uniformly bounded with respect to any $t \geq 0$ and $\nu_m(\boldsymbol{\eta}) \leq \nu_1(\boldsymbol{\eta}) = \sum_{l=1}^{\infty}\eta_l < \infty$, according to (94), we have $\int_0^{\infty}(s_m(t) - \overline{s}_m)dt > -\infty$.

(ii) The initial condition $\boldsymbol{\eta} \in \mathcal{S}$ satisfies $\eta_m \geq \overline{s}_m, \forall m \geq 0$:

According to Lemma 13, we have $s_m(t) \geq \overline{s}_m, \forall m, \forall t \geq 0$. Hence, according to equation (93), we have

$$\frac{d\nu_1(\mathbf{s}(t))}{dt} = k(\overline{s}_1 - s_1(t)) \leq 0. \tag{97}$$

Therefore, $\nu_1(\mathbf{s}(t))$ is non-increasing with respect to t. Hence, $\nu_m(\mathbf{s}(t)) \leq \nu_1(\mathbf{s}(t)) \leq \nu_1(\boldsymbol{\eta}) < \infty$, which also implies that $\nu_m(\mathbf{s}(t))$ is uniformly bounded with respect to $t \geq 0$.

Note that we have the desired result if we can show that $\int_0^{\infty}(s_m(t) - \overline{s}_m)dt < \infty$. Next, we use mathematical induction to show that

$$\int_0^{\infty}(s_m(t) - \overline{s}_m)dt < \infty \tag{98}$$

holds for any non-negative integer m.

It is obvious that (98) holds for $m = 0$. Assume that (98) holds for $m-1$. Then, we have

$$\begin{aligned}&\int_0^{\infty}\left(f(s_{m-1}(t)) - f(\overline{s}_{m-1})\right)dt \\ \overset{(a)}{=}&\int_0^{\infty} f'(z)(s_{m-1}(t) - \overline{s}_{m-1})dt \\ \overset{(b)}{\leq} n&\int_0^{\infty}(s_{m-1}(t) - \overline{s}_{m-1})dt < \infty,\end{aligned} \tag{99}$$

where the step (a) uses the Mean-Value Theorem for some z between $s_{m-1}(t)$ and $\overline{s}_{m-1}$; step (b) utilizes the boundedness of derivative of the function $f(x)$ (cf. Lemma 2). Since $\nu_m(\mathbf{s}(t))$ is uniformly bounded with respect to any $t \geq 0$ and $\nu_m(\boldsymbol{\eta}) \leq \nu_1(\boldsymbol{\eta}) = \sum_{l=1}^{\infty}\eta_l < \infty$, according to (94), we have $\int_0^{\infty}(s_m(t) - \overline{s}_m)dt < \infty$. ∎

We are ready to show Proposition 4. For any initial condition $\boldsymbol{\eta} \in \mathcal{S}$, define two initial conditions $\boldsymbol{\eta}^{(l)}$ and $\boldsymbol{\eta}^{(u)}$ satisfying $\eta_m^{(l)} \triangleq \min\{\eta_m, \overline{s}_m\}$ and $\eta_m^{(u)} \triangleq \max\{\eta_m, \overline{s}_m\}$ for any non-negative integer m. Therefore, $\eta_m^{(l)} \leq \eta_m \leq \eta_m^{(u)}, \forall m$. Let $\mathbf{s}^{(l)}(t)$ and $\mathbf{s}^{(u)}(t)$ be the solutions of the differential equations with initial conditions $\boldsymbol{\eta}^{(l)}$ and $\boldsymbol{\eta}^{(u)}$ respectively. According to Lemma 13, we have $s_m^{(l)}(t) \leq s_m(t) \leq s_m^{(u)}(t)$, $\forall t \geq 0, \forall m = 1, 2, \cdots$, which implies that

$$s_m^{(l)}(t) - \overline{s}_m \leq s_m(t) - \overline{s}_m \leq s_m^{(u)}(t) - \overline{s}_m, \forall t \geq 0, \forall m.$$

According to Lemma 14, we have

$$\lim_{t\to\infty}(s_m^{(l)}(t) - \overline{s}_m) = \lim_{t\to\infty}(s_m^{(u)}(t) - \overline{s}_m) = 0. \tag{100}$$

Therefore, we have $\lim_{t\to\infty}(s_m(t) - \overline{s}_m) = 0$.

APPENDIX I
PROOF OF PROPOSITION 5

We will omit the superscript L for brevity. First, we note that the underlying queue-length process is always symmetric under the storage scheme we mentioned in Section IV, and hence $\boldsymbol{\Phi}(t)$ is a Markov chain, . Given the current state $\boldsymbol{\Phi}(t) =$

$(\Phi_1, \Phi_2, \cdots, \Phi_m, \cdots)$, the transition rate from state $\mathbf{\Phi}$ to state $\mathbf{\Phi} + \mathbf{d}$ is given by $R_{\mathbf{d}}(\mathbf{\Phi}(t))$, where $\mathbf{d} = (d_1, d_2, \cdots, d_i, \cdots)$ and $d_i \in \{-1, 0, 1, 2, \cdots, k\}, \forall i \geq 1$. Let $\mathcal{D}$ be a collection of vectors $\mathbf{d}$. Then, Markov chain $\mathbf{\Phi}(t)$ can be written as follows:

$$\mathbf{\Phi}(t) = \mathbf{\Phi}(0) + \sum_{\mathbf{d}\in\mathcal{D}} \mathbf{d} N_{\mathbf{d}} \left(\int_0^t R_{\mathbf{d}} \left(\mathbf{\Phi}(\tau) \right) d\tau \right), \tag{101}$$

where $N_{\mathbf{d}}(t), \forall \mathbf{d} \in \mathcal{D}$, are independent Poisson processes with unit rate. Dividing by L on both sides, we obtain

$$\boldsymbol{\phi}(t) = \boldsymbol{\phi}(0) + \sum_{\mathbf{d}\in\mathcal{D}} \frac{\mathbf{d}}{L} N_{\mathbf{d}} \left(\int_0^t R_{\mathbf{d}} \left(L\boldsymbol{\phi}(\tau) \right) d\tau \right). \tag{102}$$

Let $\mathcal{H} \triangleq \{\mathbf{d} = (d_m)_{m\geq 1} \in \mathcal{D} : d_i = 0, \forall i \geq M\}$. Hence, $\boldsymbol{\phi}(t)$ can be rewritten as

$$\begin{aligned}
\boldsymbol{\phi}(t) =& \boldsymbol{\phi}(0) + \sum_{\mathbf{d}\in\mathcal{D}\setminus\mathcal{H}} \frac{\mathbf{d}}{L} N_{\mathbf{d}} \left(\int_0^t R_{\mathbf{d}} \left(L\boldsymbol{\phi}(\tau) \right) d\tau \right) + \sum_{\mathbf{d}\in\mathcal{H}} \frac{\mathbf{d}}{L} N_{\mathbf{d}} \left(\int_0^t R_{\mathbf{d}} \left(L\boldsymbol{\phi}(\tau) \right) d\tau \right) \\
=& \boldsymbol{\phi}(0) + \sum_{\mathbf{d}\in\mathcal{D}\setminus\mathcal{H}} \frac{\mathbf{d}}{L} N_{\mathbf{d}} \left(\int_0^t R_{\mathbf{d}} \left(L\boldsymbol{\phi}(\tau) \right) d\tau \right) \\
&+ \sum_{\mathbf{d}\in\mathcal{H}} \frac{\mathbf{d}}{L} \widetilde{N}_{\mathbf{d}} \left(\int_0^t R_{\mathbf{d}} \left(L\boldsymbol{\phi}(\tau) \right) d\tau \right) + \sum_{\mathbf{d}\in\mathcal{H}} \frac{\mathbf{d}}{L} \int_0^t R_{\mathbf{d}} \left(L\boldsymbol{\phi}(\tau) \right) d\tau,
\end{aligned} \tag{103}$$

where $\widetilde{N}_{\mathbf{d}}(t) \triangleq N_{\mathbf{d}}(t) - t$ denotes the centered Poisson process with unit rate.

Note that $\mathbf{s}(t) = \mathbf{s}(0) + \int_0^t \mathbf{F}^{(n,k)}(\mathbf{s}(\tau))d\tau$, where $\mathbf{F}^{(n,k)}(\mathbf{s})$ is defined in (27). Therefore, we have

$$\begin{aligned}
&\boldsymbol{\phi}(t) - \boldsymbol{s}(t) \\
=& \boldsymbol{\phi}(0) - \mathbf{s}(0) + \sum_{\mathbf{d}\in\mathcal{D}\setminus\mathcal{H}} \frac{\mathbf{d}}{L} N_{\mathbf{d}} \left(\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau \right) + \sum_{\mathbf{d}\in\mathcal{H}} \frac{\mathbf{d}}{L} \widetilde{N}_{\mathbf{d}} \left(\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau \right) \\
&+ \sum_{\mathbf{d}\in\mathcal{H}} \frac{\mathbf{d}}{L} \int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau - \int_0^t \mathbf{F}^{(n,k)}(\boldsymbol{\phi}(\tau))d\tau + \int_0^t \left(\mathbf{F}^{(n,k)}(\boldsymbol{\phi}(\tau)) - \mathbf{F}^{(n,k)}(\mathbf{s}(\tau)) \right) d\tau.
\end{aligned} \tag{104}$$

This implies

$$\begin{aligned}
&\|\boldsymbol{\phi}(t) - \boldsymbol{s}(t)\|_1 \\
\leq& \|\boldsymbol{\phi}(0) - \mathbf{s}(0)\|_1 + \left\| \sum_{\mathbf{d}\in\mathcal{D}\setminus\mathcal{H}} \frac{\mathbf{d}}{L} N_{\mathbf{d}} \left(\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau \right) \right\|_1 \\
&+ \left\| \sum_{\mathbf{d}\in\mathcal{H}} \frac{\mathbf{d}}{L} \widetilde{N}_{\mathbf{d}} \left(\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau \right) \right\|_1 \\
&+ \left\| \sum_{\mathbf{d}\in\mathcal{H}} \frac{\mathbf{d}}{L} \int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau - \int_0^t \mathbf{F}^{(n,k)}(\boldsymbol{\phi}(\tau))d\tau \right\|_1 \\
&+ \int_0^t G\|\boldsymbol{\phi}(\tau)) - \mathbf{s}(\tau)\|_1 d\tau,
\end{aligned} \tag{105}$$

where we utilize Lipschitz continuity of $\mathbf{F}^{(n,k)}(\mathbf{x})$ (cf. Lemma 9) and recall that $\|\mathbf{x}\|_1 = \sum_{m=0}^{\infty} |x_m|$ denotes the l_1 norm of

the vector $\mathbf{x} \in \mathcal{S}$. By setting $\rho(t) \triangleq \|\boldsymbol{\phi}(t) - \mathbf{s}(t)\|_1$, we have

$$\begin{aligned}
&\Pr\left\{\sup_{t\in[0,T]}\left(\rho(t) - G\int_0^t \rho(\tau)d\tau\right) \geq 4\epsilon\right\} \\
&\leq \Pr\left\{\rho(0) \geq \epsilon\right\} \\
&+ \Pr\left\{\sup_{t\in[0,T]}\left\|\sum_{\mathbf{d}\in\mathcal{D}\setminus\mathcal{H}}\frac{\mathbf{d}}{L}N_{\mathbf{d}}\left(\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau\right)\right\|_1 \geq \epsilon\right\} \qquad (106)\\
&+ \Pr\left\{\sup_{t\in[0,T]}\left\|\sum_{\mathbf{d}\in\mathcal{H}}\frac{\mathbf{d}}{L}\widetilde{N}_{\mathbf{d}}\left(\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau\right)\right\|_1 \geq \epsilon\right\} \qquad (107)\\
&+ \Pr\left\{\sup_{t\in[0,T]}\left\|\sum_{\mathbf{d}\in\mathcal{H}}\frac{\mathbf{d}}{L}\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau - \int_0^t \mathbf{F}^{(n,k)}(\boldsymbol{\phi}(\tau))d\tau\right\|_1 \geq \epsilon\right\} \qquad (108)
\end{aligned}$$

Note that $\boldsymbol{\phi}(0) \to \mathbf{s}(0)$ in probability, we have $\lim_{L\to\infty}\Pr\{\rho(0) \geq \epsilon\} = 0$. Setting $M = \frac{1}{4}\log_{k+2} L$ and combining with Lemmas 15, 16 and 17, we have

$$\lim_{L\to\infty}\Pr\left\{\sup_{t\in[0,T]}\left(\rho(t) - G\int_0^t \rho(\tau)d\tau\right) \leq 4\epsilon\right\} = 1. \qquad (109)$$

By Gronwall's Inequality, $\rho(t) - G\int_0^t \rho(\tau)d\tau \leq 4\epsilon$ implies $\rho(t) \leq 4\epsilon e^{Gt} \leq 4\epsilon e^{GT}, \forall t \in [0,T]$. Therefore, we have

$$\lim_{L\to\infty}\Pr\left\{\sup_{t\in[0,T]}\rho(t) \leq 4\epsilon e^{GT}\right\} = 1, \qquad (110)$$

which implies the desired result.

Lemma 15: Setting $M = \frac{1}{4}\log_{k+2} L$, (106) $\to 0$, as $L \to \infty$.

Proof: Let $A(T)$ and $D(T)$ be the total number of job arrivals (file access requests) and departures within the finite time interval $[0,T]$, respectively. Define event

$$\mathcal{F} \triangleq \left\{A(T) \leq (1+\alpha)L\lambda T, D(T) \leq (1+\alpha)LkT, \text{ and } \|\boldsymbol{\phi}(0)\|_1 \leq (1+\alpha)\|\mathbf{s}(0)\|_1\right\}, \qquad (111)$$

where α is a constant positive real number. It is easy to check that $\lim_{L\to\infty}\Pr\{\mathcal{F}\} = 1$.

Note that $L\|\boldsymbol{\phi}(t)\|_1$ denotes total queue lengths in the system at time t. Hence, whenever event $\mathcal{F}$ happens, we have

$$\max_{0\leq t\leq T}\|\boldsymbol{\phi}(t)\|_1 \leq (1+\alpha)\left(\lambda T + \|\mathbf{s}(0)\|_1\right) \triangleq J_\alpha < \infty.$$

In such a case, since $\phi_1(t) \geq \phi_2(t) \geq \ldots \geq \phi_M(t)$, we have

$$M\phi_M(t) \leq \sum_{m=1}^{M}\phi_m(t) \leq \max_{0\leq t\leq T}\|\boldsymbol{\phi}(t)\|_1 \leq J_\alpha, \qquad (112)$$

which implies

$$\phi_M(t) \leq \frac{J_\alpha}{M}, \forall t \in [0,T]. \qquad (113)$$

This means that when the event $\mathcal{F}$ happens, the fraction of queues with length at least M is not more than J_α/M for all $t \in [0,T]$. Therefore, we have

$$\begin{aligned}
(106) &\leq \Pr\left\{\sup_{t\in[0,T]}\frac{k}{L}N_0\left(\sum_{\mathbf{d}\in\mathcal{D}\setminus\mathcal{H}}\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau\right) \geq \epsilon\right\} \\
&\overset{(a)}{\leq} \Pr\left\{\sup_{t\in[0,T]}\frac{k}{L}N_0\left(\sum_{\mathbf{d}\in\mathcal{D}\setminus\mathcal{H}}\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau\right) \geq \epsilon; \mathcal{F}\right\} + \Pr\{\mathcal{F}^c\} \\
&\overset{(b)}{\leq} \frac{k\mathbb{E}\left[N_0\left(\sum_{\mathbf{d}\in\mathcal{D}\setminus\mathcal{H}}\int_0^T R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau\right); \mathcal{F}\right]}{L\epsilon} + \Pr\{\mathcal{F}^c\} \\
&\overset{(c)}{\leq} \frac{k(1+\alpha)(k+\lambda)TJ_\alpha}{M\epsilon} + \Pr\{\mathcal{F}^c\}, \qquad (114)
\end{aligned}$$

where the step (a) is true for $\mathcal{F}^c$ being the complement of the event $\mathcal{F}$ and $N_0(t)$ being a Poisson process with unit rate; step (b) follows from the Markov's Inequality and the monotonicity of $N_0(t)$ with respect to t; step (c) follows from the fact that the state transition only happens among queues with length at least M, the definition of the event $\mathcal{F}$, (113), and the fact that the total number of state transitions is equal to the total number of arrivals and departures. Noting that the right hand side of (114) goes to zero as $L \to \infty$ since $M = \frac{1}{4}\log_{k+2} L$, we have the desired result. ■

Lemma 16: Setting $M = \frac{1}{4}\log_{k+2} L$, (107) $\to 0$, as $L \to \infty$.

Proof: For a sufficiently large L (e.g., $L \geq k/\lambda$), we have

$$\begin{aligned}(107) &\leq \Pr\left\{\frac{k}{L}\sum_{\mathbf{d}\in\mathcal{H}}\sup_{t\in[0,T]}\left|\widetilde{N}_{\mathbf{d}}\left(\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau\right)\right| \geq \epsilon\right\}\\ &\overset{(a)}{\leq} \Pr\left\{\sup_{t\in[0,T]}\left\|\widetilde{N}_{\mathbf{d}^*}\left(\int_0^t R_{\mathbf{d}^*}(L\boldsymbol{\phi}(\tau))d\tau\right)\right\| \geq \frac{L\epsilon}{k(k+2)^M}\right\}\\ &\overset{(b)}{\leq} \Pr\left\{\sup_{t\in[0,L\lambda T]}\left\|\widetilde{N}_{\mathbf{d}^*}(t)\right\| \geq \frac{L\epsilon}{k(k+2)^M}\right\}\\ &\overset{(c)}{\leq} 2\exp\left(-L\lambda T\cdot\sigma\left(\frac{\epsilon}{k(k+2)^M\lambda T}\right)\right),\end{aligned} \tag{115}$$

where the step (a) is true for

$$\mathbf{d}^* \in \arg\max_{\mathbf{d}\in\mathcal{H}}\sup_{t\in[0,T]}\left|\widetilde{N}_{\mathbf{d}}\left(\int_0^t R_{\mathbf{d}}(L\boldsymbol{\phi}(\tau))d\tau\right)\right|,$$

and utilizes the fact that there are at most $(k+2)^M$ different vectors within the finite space $\mathcal{H}$; step (b) follows from the fact that if $\gamma(t) \leq Ct$ for some constant $C > 0$, then

$$\sup_{t\in[0,T]}\nu(\gamma(t)) \leq \sup_{t\in[0,CT]}\nu(t)(\text{since } \nu(\gamma(t)) \leq \sup_{s\in[0,Ct]}\nu(s)),$$

and the fact that $R_{\mathbf{d}}(L\boldsymbol{\phi}) \leq \max\{L\lambda, k\} = L\lambda$ for $L \geq k/\lambda$; step (c) is true for $\sigma(t) = (1+t)\log(1+t) - t$, and utilizes [28, Proposition 5.2].

Noting that $\sigma(0) = \sigma'(0) = 0$ and $\sigma''(0) = 1$, we have $\sigma(t) = \frac{1}{2}t^2$ for sufficiently small t according to Taylor Expansion. Hence, for a sufficiently large L, we have

$$(107) \leq 2\exp\left(-\frac{L\epsilon^2}{2k^2(k+2)^{2M}\lambda T}\right). \tag{116}$$

Noting that $M = \frac{1}{4}\log_{k+2} L$, we have the desired result. ■

Lemma 17: Setting $M = \frac{1}{4}\log_{k+2} L$, (108) $\to 0$, as $L \to \infty$.

Proof: We first note that

$$\begin{aligned}&\left\|\sum_{\mathbf{d}\in\mathcal{H}}\frac{\mathbf{d}}{L}\int_0^t R_d(L\boldsymbol{\phi}(\tau))d\tau - \int_0^t \mathbf{F}^{(n,k)}(\boldsymbol{\phi}(\tau))d\tau\right\|_1\\ &\overset{(a)}{\leq}\int_0^t\left\|\mathbf{F}^{(L)}(\boldsymbol{\phi}(\tau)) - \mathbf{F}^{(n,k)}(\boldsymbol{\phi}(\tau))\right\|_1 d\tau\\ &\overset{(b)}{=}\int_0^t\sum_{m=M+1}^{\infty}\left|F_m^{(n,k)}(\boldsymbol{\phi}(\tau))\right|d\tau,\end{aligned} \tag{117}$$

where the step (a) is true for $\mathbf{F}^{(L)}(\boldsymbol{\phi}) \triangleq \sum_{\mathbf{d}\in\mathcal{H}}\frac{\mathbf{d}}{L}R_{\mathbf{d}}(L\boldsymbol{\phi})$ and uses the fact that

$$\left\|\int_0^t(\mathbf{X}(\tau) - \mathbf{Y}(\tau))d\tau\right\|_1 \leq \int_0^t\|\mathbf{X}(\tau) - \mathbf{Y}(\tau)\|_1 d\tau;$$

step (b) follows from the fact that $F_m^{(L)}(\boldsymbol{\phi}) = F_m^{(n,k)}(\boldsymbol{\phi}) = \lambda\left(f^{(n,k)}(\phi_{m-1}) - f^{(n,k)}(\phi_m)\right) - k(\phi_m - \phi_{m+1})$ for any $m \leq M$ and $F_m^{(L)}(\boldsymbol{\phi}) = 0$ for any $m \geq M+1$ due to the fact that $d_m = 0, \forall m \geq M+1$ for any $\mathbf{d} \in \mathcal{H}$.

According to the definition of the $F_m^{(n,k)}(\boldsymbol{\phi})$, we have

$$\begin{aligned}\left|F_m^{(n,k)}(\boldsymbol{\phi})\right| &\overset{(a)}{\leq} \lambda\left(f^{(n,k)}(\phi_{m-1}) - f^{(n,k)}(\phi_m)\right) + k\left(\phi_m - \phi_{m+1}\right)\\ &\overset{(b)}{\leq} n\lambda\left(\phi_{m-1} - \phi_m\right) + k\left(\phi_m - \phi_{m+1}\right),\end{aligned} \tag{118}$$

where the step (a) is true uses the fact that $\phi_{m-1} \geq \phi_m \geq \phi_{m+1}$ and the non-decreasing property of the function $f^{(n,k)}(t)$ (cf. Lemma 2); step (b) uses Mean-Value Theorem and the fact that $0 \leq \left(f^{(n,k)}(x)\right)' \leq n$ (cf. Lemma 2).

By substituting (118) into (117), we have

$$\begin{aligned}&\left\|\sum_{\mathbf{d}\in\mathcal{H}} \frac{\mathbf{d}}{L}\int_0^t R_d(L\boldsymbol{\phi}(\tau))d\tau - \int_0^t \mathbf{F}^{(n,k)}(\boldsymbol{\phi}(\tau))d\tau\right\|_1\\ &\leq \int_0^t (n\lambda\phi_M(\tau) + k\phi_{M+1}(\tau))d\tau,\end{aligned} \tag{119}$$

where we use the fact that $\sum_{m=M}^{\infty}\phi_m < \infty$.

Therefore, we have

$$\begin{aligned}(108) &\leq \Pr\left\{\int_0^T \left(n\lambda\phi_M(\tau) + k\phi_{M+1}(\tau)\right)d\tau \geq \epsilon; \mathcal{F}\right\} + \Pr\{\mathcal{F}^c\}\\ &\leq \mathbb{1}_{\left\{\frac{(n\lambda+k)J_\alpha T}{M} \geq \epsilon\right\}} + \Pr\{\mathcal{F}^c\},\end{aligned} \tag{120}$$

where the last step uses the fact that state transitions only happen among queues with length at least M, the definition of $\mathcal{F}$, and (113). Noting that $M = \frac{1}{4}\log_{k+2} L$, we have the desired result. ∎

REFERENCES

[1] B. Li, A. Ramamoorthy, and R. Srikant, "Mean-field-analysis of coding versus replication in cloud storage systems," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, April 2016.

[2] S. Lin and D. J. Costello, *Error Control Coding, 2nd Ed.* Prentice Hall, 2004.

[3] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," in *Proc. ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, Philadelphia, PA, USA, August 2005.

[4] G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica, "Why let resources idle? aggressive cloning of jobs with dolly," in *Proc. USENIX Conference on Hot Topics in Cloud Ccomputing (HotCloud)*, Boston, MA, USA, June 2012.

[5] A. Vulimiri, P. B. Godfrey, R. Mittal, J. Sherry, S. Ratnasamy, and S. Shenker, "Low latency via redundancy," in *Proc. ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, Santa Barbara, CA, USA, December 2013.

[6] L. Huang, S. Pawar, H. Zhang, and K. Ramchandran, "Codes can reduce queueing delay in data centers," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Cambridge, MA, USA, July 2012.

[7] N. B. Shah, K. Lee, and K. Ramchandran, "The MDS queue: Analysing the latency performance of erasure codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Honolulu, HI, USA, July 2014.

[8] ——, "When do redundant requests reduce latency?" in *Proc. Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, USA, October 2013.

[9] G. Joshi, Y. Liu, and E. Soljanin, "On the delay-storage trade-off in content download from coded distributed storage systems," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 989–997, 2014.

[10] Y. Xiang, T. Lan, V. Aggarwal, and Y. F. R. Chen, "Joint latency and cost optimization for erasurecoded data center storage," *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 2, pp. 3–14, 2014.

[11] S. Chen, Y. Sun, U. C. Kozat, L. Huang, P. Sinha, G. Liang, X. Liu, and N. B. Shroff, "When queueing meets coding: Optimal-latency data retrieving scheme in storage clouds," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Toronto, Canada, April 2014.

[12] G. Liang and U. C. Kozat, "TOFEC: Achieving optimal throughput-delay trade-off of cloud storage using erasure codes," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Toronto, Canada, April 2014.

[13] Y. Sun, Z. Zheng, C. E. Koksal, K. Kim, and N. B. Shroff, "Probably delay efficient data retrieving in storage clouds," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Hong Kong, China, April 2015.

[14] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, E. Hyytia, and A. Scheller-Wolf., "Reducing latency via redundant requests: Exact analysis," in *Proc. ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Portland, OR, USA, June 2015.

[15] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich, "Queueing system with selection of the shortest of two queues: An asymptotic approach," *Problemy Peredachi Informatsii*, vol. 32, no. 1, pp. 20–34, 1996.

[16] M. Mitzenmacher, *The power of two choices in randomized load balancing*. Ph.D. Thesis, University of California at Berkeley, 1996.

[17] M. Bramson, Y. Lu, and B. Prabhakar, "Randomized load balancing with general service time distributions," in *Proc. ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, New York, NY, USA, June 2010.

[18] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. Greenberg, "Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services," *Performance Evaluation*, vol. 68, no. 11, pp. 1056–1071, 2011.

[19] L. Ying, R. Srikant, and X. Kang, "The power of slightly more than one sample in randomized load balancing," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Hong Kong, April 2015.

[20] A. L. Stolyar, "Pull-based load distribution in large-scale heterogeneous service systems," *Queueing Systems*, vol. 80, no. 11, pp. 341–361, 2015.

[21] K. Gardner, M. Harchol-Balter, M. V. A. Scheller-Wolf, and S. Zbarsky, "Redundancy-d: The power of d choices for redundancy," *To appear in Operations Research*, 2016.

[22] K. Ousterhout, P. Wendell, M. Zaharia, and I. Stoica, "Sparrow: distributed, low latency scheduling," in *Proc. ACM Symposium on Operating Systems Principles (SOSP)*, Pennsylvania, PA, USA, November 2013.

[23] M. Lugo, *A Note for Stat 134 Fall 2011: The Expectation of the Maximum of Exponentials*. University of California at Berkeley, 2011.

[24] A. Weiss and A. Shwartz, "Large deviations for performance analysis," 1995.
[25] R. Srikant and L. Ying, *Communication Networks: An Optimization, Control, and Stochastic Networks Perspective*. Cambridge University Press, 2013.
[26] B. Hajek, *Notes for ECE 467: Communication Network Analysis*. University of Illinois at Urbana-Champaign, 2006.
[27] D. E. Knuth, R. L. Graham, O. Patashnik *et al.*, "Concrete mathematics," *Adison Wesley*, 1989.
[28] M. Draief and L. Massouli, *Epidemics and rumours in complex networks*. Cambridge University Press, 2010.