Counterintuitive Characteristics of Optimal Distributed LRU Caching Over Unreliable Channels

Guocong Quan, Jian Tan, and Atilla Eryilmaz, *Senior Member, IEEE*

***Abstract*—Least-recently-used (LRU) caching and its variants have conventionally been used as a fundamental and critical method to ensure fast and efficient data access in computer and communication systems. Emerging data-intensive applications over unreliable channels, e.g., mobile edge computing and wireless content delivery networks, have imposed new challenges in optimizing LRU caching in environments prone to failures. Most existing studies focus on reliable channels, e.g., on wired Web servers and within data centers, which have already yielded good insights and successful algorithms. Surprisingly, we show that these insights do not necessarily hold true for unreliable channels. We consider a single-hop multi-cache distributed system with data items being dispatched by random hashing. The objective is to design efficient cache organization and data placement that minimize the miss probability. The former allocates the total memory space to each of the involved caches. The latter decides data routing and replication strategies. Analytically, we characterize the asymptotic miss probabilities for unreliable LRU caches, and optimize the system design. Remarkably, these results sometimes are counterintuitive, differing from the ones obtained for reliable caches. We discover an interesting phenomenon: allocating the cache space unequally can achieve a better performance, even when channel reliability levels are equal. In addition, we prove that splitting the total cache space into separate LRU caches can achieve a lower asymptotic miss probability than organizing the total space in a single LRU cache. These results provide new and even counterintuitive insights that motivate novel designs for caching systems over unreliable channels.**

***Index Terms*—LRU caching, reliability, memory space allocation.**

I. Introduction

Caching is a fundamental and critical method in modern computer and communication systems that can efficiently accelerate data access [1], [2], [3], [4], [5] at the expense of a dedicated fast memory space. As default algorithms, the least-recently-used (LRU) caching and its variants [6], [7], [8] have been predominantly used to manage the allocated cache space in various computer systems [3], [4], [9]. Under the LRU algorithm, only the most recently used data items are stored in the cache. If the cache is full and the requested data cannot be found therein, the data item that has not been used for the longest time will be moved out of the cache to make room for the newly requested one. To design efficient caching systems that minimize the miss ratios, a fundamental problem is to optimize cache organization and data placement. The former decides the optimal memory allocation to the involved caches and the latter dispatches data requests to the right caches.

The existing work on cache optimization almost focuses on improving the performance over reliable channels, e.g., within data centers and on wired Web servers [10], [11], [12], [13]. These studies have yielded good insights with successful algorithms in real systems [3], [14]. However, with the increasing popularity of emerging data applications over wireless and mobile networks, e.g., mobile edge computing and content delivery networks, cached data delivered over unreliable channels become substantial in data-intensive applications [15], [16], [17], [18]. Due to mobility, fading, communication errors, etc, the access to caches could fail intermittently or be significantly delayed. Consequently, a high fetching cost can be incurred on unreliable channels even when the requested data items are indeed in the cache. This fact is significantly different from reliable channels. Thus, caching the same data item in multiple caches, i.e., data replications, should almost always be considered in presence of channel failures. All these features impose new challenges in optimizing cache organization and data placement in environments prone to failures. It merits a deeper investigation on whether the insights and engineering practices that are optimized for reliable caches can still work well in unreliable environments. If not, what to change?

We consider a single-hop multi-cache distributed system with data items being dispatched through random hashing to multiple caches. One important decision is to route the data items to the right cache. The current practice unanimously relies on an effective scale-out method that is called consistent hashing [19]. Under consistent hashing, data requests are routed to different caches according to a hash function. To be general and analytically tractable for our modeling analysis, we consider a dispatching scheme based on hashing that satisfies the Simple Uniform Hashing Assumption (SUHA). To support data replication for unreliable channels, we assume the random hash function maps each data item to a set of caches instead of a single one. Then, we optimize the data replications by carefully designing the hash functions and the cache space allocation. In particular, we make the following contributions to the literature on LRU caching:

- We propose a tractable model for LRU caching over unreliable channels, which considers cache organization and data placement simultaneously. More importantly, we derive the asymptotic miss probability in a closed form, which provides an effective tool to optimize caching system performance.
- We characterize the property of the optimal cache space

Manuscript received December 10, 2019; revised March 5, 2020; accepted July 8, 2020. This work was supported by NSF grants CNS-NeTS-1514260, CNS-NeTS-1717045, CNS-NeTS-1717060, CNS-NeTS-2007231, CMMI-SMOR-1562065, CNS-ICN-WEN-1719371, and CNS-SpecEES-1824337, ONR Grant N00014-19-1-2621, and DTRA grant HDTRA1-18-1-0050

The authors are with The Ohio State University, Columbus, OH 43210, USA (emails: {quan.72, tan.252, eryilmaz.2}@osu.edu).

allocation and the data replication. Surprisingly, the results are quite different from those for reliable channels. A counterintuitive phenomenon is discovered: allocating the cache spaces unequally is better than the equal arrangement even when the channels have identical reliability levels. Moreover, we propose an explicit non-identical separation policy that outperforms the identical separation policy. We also generalize the results for channels with heterogenous reliability levels. These new insights deepen our understanding of LRU caching over unreliable channels and can potentially be applied in real practice to further improve the system performance.

Related work: For reliable LRU caches, the miss probability can be accurately approximated when the cache space is large [10], [20], [21], [22], [23], [24], [25], [26]. Extensive studies have been done to optimize cache space allocation with different objectives. In order to maximize the hit ratio based utility functions, it is proved in [11] that splitting the total memory space into separate LRU caches is at least as good as resource pooling which allocates the whole memory space to a single LRU cache. A complex version of this problem considering routing decisions in reliable environments is investigated in [13]. In addition, when data sizes, popularity distributions, request rates and data overlaps are considered jointly, complex results are obtained such that cache space separation can be asymptotically better than, equal to or worse than resource pooling depending on the afore-mentioned four factors [10]. For caching over unreliable channels, most existing studies focus on optimizing data placement to improve caching gains [17], [27], [28]. In [29], it is shown that the ubiquitous path replication algorithm combined with LRU replacement policy is suboptimal in caching networks, and novel adaptive algorithms with optimality guarantees are proposed to decide which data item should be stored in the cache. However, few existing work considers cache space allocation and data placement simultaneously in unreliable environments. How to allocate cache space and dispatch data requests for LRU caching over unreliable channels still remains unexplored and deserves a thorough investigation. The conference version of this paper [30] was published in IEEE INFOCOM 2019.

II. Model Description

Consider a set of infinite data items $\mathcal{D} = \{d_1, d_2, \cdots\}$ and a data flow which is a sequence of data requests on the data set $\mathcal{D}$. Assume the size of each data item is identical and normalized to 1. Assume the requests arrive according to a Poisson process. Let $\{\tau_n, -\infty < n < +\infty\}$ denote the time points that the requests arrive. Define R_n as the data item that is requested at time τ_n, $R_n \in \mathcal{D}$. Assume R_n's are i.i.d random variables and define $\mathbb{P}[R_n = d_i] = q_i$, $i \geq 1$ as the popularity distribution. Empirical studies on real data traces have shown that the popularities often follow a Zipf's distribution. Therefore, we assume

$$q_i \sim c/i^{\alpha}, \alpha > 1, i \geq 1.$$

Note that $f(z) \sim g(z)$ means $\lim_{z\to\infty} f(z)/g(z) = 1$. To characterize the miss ratio of the system, it is sufficient to focus on the request at one time point saying τ_0 when the system reaches stationarity, because the requests are assumed to be independent. Consider a set of M LRU caches $\mathcal{C} = \{C_m : 1 \leq m \leq M\}$. To model the channel unreliability, we assume that the cache C_m can be accessed (i.e., the corresponding channel is reliable) with a probability p independently at time τ_n. The probability p represents the channel reliability level. Define a hash function $H : \mathcal{D} \to 2^{\mathcal{C}} \setminus \emptyset$ which hashes a data item to a nonempty subset of all M caches. For example, in Fig. 1, the data request d_2 is hashed to $\{C_1, C_{M-1}\}$, but only C_1 can be accessed at that time. Let N_i denote the number of elements in the set $H(d_i)$, $i \geq 1$. We assume $H(\cdot)$ is randomly selected from a set of hash functions $\mathcal{H} = \{h_w(\cdot), w = 1, 2, \cdots\}$ that satisfies Assumption 1 (i.e., SUHA property). Note that although $H(\cdot)$ is random, once a hash function $h_w(\cdot)$ is selected, each data item will be mapped to a subset of caches deterministically.

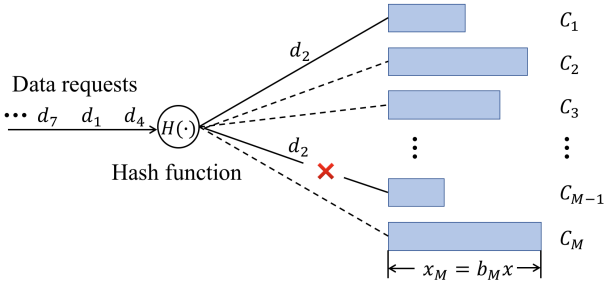

Fig. 1. Distributed caching over unreliable channels.

Assumption 1 (SUHA)**.** *Assume N_i's are i.i.d. random variables with $\mathbb{P}[N_i = m] = \mu_m$, $1 \leq m \leq M$, $\sum_{m=1}^{M} \mu_m = 1$. Given N_i, assume $H(d_i)$'s are randomly chosen from all $\binom{M}{N_i}$ possible subsets of $\mathcal{C}$ that contain N_i caches with an equal probability, i.e., $1/\binom{M}{N_i}$.*

Let $\mathcal{I}_n \triangleq H(R_n)$, $\mathcal{I}_n \subseteq \mathcal{C}$. Let $\mathcal{J}_n = \{C_m : C_m \in \mathcal{I}_n, C_m \text{ is accessible at } \tau_n, 1 \leq m \leq M\}$. When the request arrives at time τ_n, the system will fetch the data item R_n from the caches in the set $\mathcal{J}_n$. A cache hit occurs if and only if R_n is stored in at least one cache of $\mathcal{J}_n$. Otherwise, we call it a miss. Only the caches in $\mathcal{J}_n$ will be updated by the request R_n according to the LRU algorithm. We assume that the updating process can be completed as long as the cache is accessible when the request arrives. Let x denote the total cache space and $x_m = b_m x$ denote the space of cache C_m, $0 < b_m < 1$, $\sum_{m=1}^{M} b_m = 1$, $1 \leq m \leq M$. Without loss of generality, we assume the caches are sorted such that b_m is non-increasing with m. The objective of this paper is to characterize the optimal hashing mechanism $\vec{\mu}^*$ (see Assumption 1) and cache space allocation $\vec{b}^*$ under different settings.

III. Counterintuitive Insights from Analysis

The performance of reliable LRU caching ($p = 1$) has been investigated for a long time. Consider a single LRU cache with a cache space x and data items following Zipf's popularity distributions (i.e., $q_i \sim c/i^{\alpha}$, $\alpha > 1$). According to Theorem 3 of [21], we have, as $x \to \infty$

$$\mathbb{P}[\text{Miss for } R_0] \sim \frac{\Gamma(1-1/\alpha)^{\alpha}}{\alpha} \frac{c}{x^{\alpha-1}} \triangleq Q(x), \qquad (1)$$

where $\Gamma(1-1/\alpha)=\int_0^\infty t^{-1/\alpha}e^{-t}dt$ is the gamma function. For multiple LRU caches over reliable channels organized by the hashing mechanism described in Section II, the miss probability of the system can be accurately approximated by [26]. In Lemma 1, we show that, for reliable channels, pooling the total cache space into a single LRU cache can achieve a better asymptotic miss probability than splitting the cache space to multiple caches.

Lemma 1. *Consider M LRU caches organized by the hashing mechanism described in Section II. If $p=1$, then as $x\to\infty$, we have almost surely for all H,*

$$\mathbb{P}[\textit{Miss for } R_0|H] \gtrsim Q(x),$$

where the equality holds asymptotically if and only if $b_m = 1/M$, $1\le m\le M$, and $Q(x)$ is defined in Equation (1).

Note that $f(z)\gtrsim g(z)$ means $\lim_{z\to\infty} f(z)/g(z)\ge 1$. The proof is provided in Section VIII-A. Lemma 1 implies two insights for LRU caching over reliable channels:

- The asymptotic miss ratio under resource pooling is always better than or equal to that under resource separation.
- Allocating cache space unequally will achieve a worse asymptotic miss probability than allocating equally.

Interestingly, when channels are unreliable (i.e., $p<1$), these insights will not hold. In Section IV, we rigorously prove the following counterintuitive results for distributed LRU caching over unreliable channels, all building on Theorem 1.

- *When the cache spaces are required to be the same, splitting the total memory space into multiple LRU caches can achieve a better miss probability than resource pooling (cf. Theorem 2). We further characterize the splitting required to minimize the miss probability as a function of the channel reliability level and the total available memory space (cf. Theorem 3).*
- *When the cache spaces are allowed to be different, allocating the total memory space unequally to the caches can achieve a better miss probability than the equal cache space allocation, even when the reliability level is identical for each channel (cf. Theorem 4). We further develop a cache allocation policy that can yield significant improvements on the miss probability compared to the equal cache space allocation (cf. Theorem 5).*

These contradictory results for reliable and unreliable channels indicate the importance to consider channel reliability when organizing caching systems. They reveal that previously successful engineering methods for optimizing caches over reliable channels may not work well in unreliable environments. Fortunately, new insights are provided in this paper and can be potentially exploited to modify existing algorithms and further improve the system performance.

IV. Performance Analysis

In this section, we first derive the asymptotic miss probability for the distributed LRU caching over unreliable channels modeled in Section II, and then characterize the optimal cache space allocation and hashing mechanism with the objective to minimize the miss probability.

A. Miss probability

In this section, we derive accurate approximations for the cache miss probability. Given a set $\mathcal{C}$ of M caches, there are $\binom{M}{m}$ different subsets that contain m caches, $1\le m\le M$. Let $\mathcal{S}_k^{(m)}$, $1\le k\le \binom{M}{m}$ denote the subsets that contain m caches, and $I_{k,i}^{(m)}$, $1\le i\le m$ be the indices of the caches in set $\mathcal{S}_k^{(m)}$. Assume that $I_{k,i}^{(m)}$ are sorted as an increasing sequence with i. For example, if $\mathcal{S}_k^{(3)}=\{C_9,C_4,C_{10}\}$, then $(I_{k,1}^{(3)},I_{k,2}^{(3)},I_{k,3}^{(3)})=(4,9,10)$. Define $W_k^{(m)}\triangleq\mathbb{P}[\mathcal{I}_0=\mathcal{S}_k^{(m)}|H]=\sum_{i\in\{i:H(d_i)=\mathcal{S}_k^{(m)}\}} q_i$. We derive the miss probability in Theorem 1.

Theorem 1. *Under the model described in Section II, as the total cache space $x\to\infty$, we have almost surely for all H,*

$$\mathbb{P}[\textit{Miss for } R_0|H] - \sum_{m=1}^{M}\sum_{k=1}^{\binom{M}{m}}(1-p)^m W_k^{(m)} \sim \left(\sum_{m=1}^{M} L\left(m,\vec{b}\right)\mu_m\right)\left(\sum_{m=1}^{M} m\mu_m\right)^{\alpha-1} Q(x),$$

and

$$\mathbb{P}[\textit{Miss for } R_0] - \sum_{m=1}^{M}(1-p)^m\mu_m \sim \left(\sum_{m=1}^{M} L\left(m,\vec{b}\right)\mu_m\right)\left(\sum_{m=1}^{M} m\mu_m\right)^{\alpha-1} Q(x),$$

where $Q(x)$ is defined in (1) and

$$L\left(m,\vec{b}\right)=\frac{1}{\binom{M}{m}}\sum_{k=1}^{\binom{M}{m}}\sum_{(j,l):\mathcal{S}_j^{(l)}\subseteq\mathcal{S}_k^{(m)}}\left(p^l(1-p)^{m-l}\cdot\left(\sum_{i=1}^{l}(1-p)^{i-1}\left(Mb_{I_{j,i}^{(l)}}\right)^{\alpha}\right)^{1/\alpha-1}\right).$$

The proof is presented in Section VIII-B. For a given total memory space x, a hashing mechanism $\vec{\mu}$ and a space allocation strategy $\vec{b}$, the asymptotic miss probability can be explicitly approximated by the function

$$P(x;\vec{\mu},\vec{b})\triangleq\left(\sum_{m=1}^{M} L\left(m,\vec{b}\right)\mu_m\right)\left(\sum_{m=1}^{M} m\mu_m\right)^{\alpha-1} Q(x) + \sum_{m=1}^{M}(1-p)^m\mu_m. \tag{2}$$

With this effective tool, next we characterize the optimal LRU caching policy over unreliable channels. Specifically, for a given total cache space x, let $\vec{\mu}^*(x)$, $\vec{b}^*(x)$ denote the optimal hashing mechanism and the optimal cache space allocation that minimize the miss probability. We aim to characterize the asymptotic behavior of the optimal solutions, i.e., $\lim_{x\to\infty}\vec{\mu}^*(x)$ and $\lim_{x\to\infty}\vec{b}^*(x)$.

B. Equal cache space allocation

In this section, assuming the total memory space is equally allocated to each cache, i.e., $b_1 = b_2 = \cdots = b_M = 1/M$, we optimize the hashing mechanism $\vec{\mu}$ as well as the cache space allocation by determining the number of caches M.

Applying Theorem 1, we derive the asymptotic miss probability for equal cache space allocation in Corollary 1.

Corollary 1. *For $b_1 = b_2 = \cdots = b_M = 1/M$, as the total caches space $x \to \infty$, we have almost surely for all H,*

$$\mathbb{P}[\textit{Miss for } R_0] - \sum_{m=1}^{M}(1-p)^m\mu_m \sim \left(\sum_{m=1}^{M} L(m,\vec{b})\mu_m\right)\left(\sum_{m=1}^{M} m\mu_m\right)^{\alpha-1} Q(x)$$

where

$$L(m,\vec{b}) = \sum_{i=1}^{m}\binom{m}{i}p^i(1-p)^{m-i}\left(\frac{p}{1-(1-p)^i}\right)^{1-1/\alpha}.$$

Next, assuming the number of caches M is finite and fixed, we optimize the hashing mechanism by considering the following optimization problem.

$$\begin{aligned} \min_{\vec{\mu}} \quad & P\left(x;\vec{\mu},(1/M,\cdots,1/M)\right) \\ \text{subject to} \quad & \sum_{m=1}^{M}\mu_m = 1, \\ & 0 \le \mu_m \le 1, \qquad m = 1,2,\cdots,M. \end{aligned} \tag{3}$$

Let $\vec{\mu}^*(x)$ denote the optimal solution to Problem (3) for a given total cache space x. We will show that hashing the request to all caches is asymptotically optimal.

Theorem 2. *The optimal solution to Problem (3) satisfies*

$$\lim_{x\to\infty}\mu_m^*(x) = 0 \quad \textit{for } 1 \le m \le M-1,$$
$$\lim_{x\to\infty}\mu_M^*(x) = 1.$$

The proof of Theorem 2 is presented in Section VIII-C. For sufficiently large cache space, simply hashing the data item to all M caches can achieve the optimal miss probability. This should hold for any cache space allocation, because we have $\lim_{x\to\infty} Q(x) = 0$, and the miss probability approximated in (2) will be dominated by the term $\sum_{m=1}^{M}(1-p)^m\mu_m$. For the sake of rigor, in this paper, we investigate the optimal caching in the asymptotic regime, where the optimal hashing is simple as shown in Theorem 2. However, if the total cache size is small, simply hashing the request to all caches may not necessarily be optimal. Moreover, simulations in Section VI verify that the miss probability approximated by (2) is accurate for small cache sizes. How to leverage (2) to optimize caching performance for small caches deserves a separate investigation.

We define two policies as:

Resource Pooling (RP) Policy: Allocate the total cache space to a single LRU cache, and dispatch all data requests to this cache.

Equal Allocation (EA) Policy: Given $M \ge 2$, set $b_m = 1/M$ for $1 \le m \le M$, $\mu_m = 0$ for $1 \le m \le M-1$ and $\mu_M = 1$.

Let $\mathbb{P}_{\text{miss}}^{\text{RP}}$ and $\mathbb{P}_{\text{miss}}^{\text{EA}}$ denote the miss probability under the RP policy and the EA policy, respectively. We have $\lim_{x\to\infty}\mathbb{P}_{\text{miss}}^{\text{RP}} = 1-p$ and $\lim_{x\to\infty}\mathbb{P}_{\text{miss}}^{\text{EA}} = (1-p)^M$. As the total cache space goes to infinity, the miss ratio of RP and EA policies will converge to the probability that all channels fail when the request arrives. Consequently, compared to resource pooling, allocating the total cache space to multiple caches can reduce the miss probability dramatically. This conclusion is contradictory to the results for reliable caches in Lemma 1, where the asymptotic miss probability achieved by resource pooling is always smaller than or equal to that achieved by resource separation. In addition, the limiting miss probability of the EA policy decreases exponentially with the number of caches M. Does it indicates that a larger number of caches can always guarantee a better miss probability for a given total memory space? The answer is no. Let $M^*(x)$ denote the optimal number of caches for the EA policy given the total cache space x. We characterize the limiting behavior of $M^*(x)$ in the following theorem.

Theorem 3. *Assuming the data items are hashed to all M caches and $\lim_{x\to\infty} x/M = \infty$, we have, as $x \to \infty$*

$$M^*(x) \sim \frac{1-\alpha}{\log(1-p)}\log x.$$

The proof is presented in Section VIII-D. When the number of caches increases, although the probability that channels fail decreases, a miss will be more likely to happen in each cache. The optimal cache number $M^*(x)$ balances this trade-off. Theorem 3, in conjunction with Theorem 2, implies that the miss probability under the EA policy tends to zero for large x, which is better than the RP policy that will always have a positive lower limit (i.e., $1-p$).

C. Unequal cache space allocation

In this section, we answer the following question. Given a total memory space and M caches, if the cache spaces are allowed to be unequal, can we achieve a better miss probability than allocating the total memory space equally? For LRU caching over reliable channels ($p = 1$), it is shown in Lemma 1 that $b_m = 1/M$, $1 \le m \le M$ is the optimal solution. Will this result still hold when channels are not reliable ($p < 1$)? In this section, we show that if $p < 1$, choosing b_m's unequally can further reduce the asymptotic miss probability.

For a fixed M and any given space allocation method $\vec{b}$ satisfying $b_m \ne 0$ for $\forall 1 \le m \le M$, similar to the equal allocation case (Theorem 2), the asymptotically optimal hashing mechanism is $\mu_M^* = 1$, $\mu_m^* = 0$, $1 \le m \le M-1$, when the total memory space x is large enough. To minimize $P(x;\vec{\mu}^*,\vec{b})$ which is defined in (2), we formulate the following problem.

$$\begin{aligned} \min_{\vec{b}} \quad & P\left(x;(0,\cdots,0,1),\vec{b}\right) \\ \text{subject to} \quad & \sum_{m=1}^{M} b_m = 1, \\ & 0 \le b_m \le 1, \qquad m = 1,2,\cdots,M. \end{aligned} \tag{4}$$

It can be verified that Problem (4) is nonconvex. Finding the global optimum $\vec{b}^*(x)$ still remains an open problem. However, we prove that allocating cache space equally is not the optimal solution and provide an easy-to-implement policy to improve the performance of equal allocation.

Theorem 4. *For M caches with $p \in (0,1)$, the optimal solution to Problem (4) satisfies $b_i^*(x) > b_j^*(x)$ for $\forall 1 \le i < j \le M$.*

See Section VIII-E for the proof. Theorem 4 shows a counterintuitive result that unequal cache space allocation can achieve a better miss probability than equal allocation, even when the unreliable probability p is identical for each channel. To understand why equal allocation is not the optimal, we investigate the optimal static caching policy. Under a static policy, the popularity of each data item is pre-known, based on which the cache space allocation and the data placement are designed. For reliable channels, the optimal static policy stores the most popular data items in one of the caches. For unreliable caches, however, the static optimal policy is nontrivial due to the potential benefits of data replications. Let x_m°, $1 \le m \le M$, denote the memory space allocated to the cache C_m under the optimal static policy. Define $b_m^\circ = x_m^\circ / \sum_{i=1}^M x_i^\circ, 1 \le m \le M$. The solution of x_m°, $1 \le m \le M$, is not unique. In the following lemma, we present one optimal static policy for unreliable channels. In order to have compact descriptions, we assume that x_m°'s are non-negative integers in the following lemma.

Lemma 2 (An Optimal Static Policy)**.** *If the popularity of each content were known a priori, the following static policy minimizes the miss probability for M caches with a total memory space x:*
Cache space allocation: Set $x_1^\circ \ge x_2^\circ \ge \cdots \ge x_M^\circ$, satisfying

$$\frac{(1-p)^{j-1}}{(x_j^\circ)^\alpha} \ge \frac{(1-p)^{k-1}}{(x_k^\circ+1)^\alpha} \quad \text{for } \forall x_j^\circ \ge 1, x_k^\circ \ge 1,$$
$$\sum_{m=1}^{M} x_m^\circ = \lfloor x \rfloor;$$

Data placement: Store data items $\{d_i : 1 \le i \le x_m^\circ\}$ in cache C_m, if $x_m^\circ \ge 1$.

A proof of this lemma is presented in Section VIII-F. As the cache space goes to infinity, the optimal static allocation can be explicitly calculated as

$$\lim_{x\to\infty} b_m^\circ(x) = (1-p)^{(m-1)/\alpha} \frac{1-(1-p)^{1/\alpha}}{1-(1-p)^{M/\alpha}}. \quad (5)$$

Based on Lemma 2, we summarize the following key insights, which intuitively explain why unequal allocation can achieve better miss ratios than equal allocation.

Key Insights: The above optimal static policy explicitly characterizes how many caches that each item must be stored in. While the most popular items in the set $\{d_i : 1 \le i \le \lfloor x_M^\circ \rfloor\}$ are stored in all M caches, progressively less popular items are stored in a decreasing number of caches, as described in Lemma 2. As such, this policy optimally balances the trade-off between the cost of storing the same item in multiple caches and the likelihood of finding a requested item in at least one of the connected caches. Despite its optimality guarantee, the optimal static policy is not directly implementable since it requires the knowledge of the popularity of each item to determine the cache allocation and data placement. However, it provides useful insights for the design of dynamic cache management policies where the popularity of the items are unknown. In particular, by allocating cache space *unequally*, as dictated by the static design, and by using LRU cache management at each of the cache, which adaptively maintains more popular requests in its cache, we can obtain a counter-intuitive design of a dynamic *unequal* caching policy over unreliable channels.

Next, we propose an unequal cache space allocation policy that significantly improves the performance of equal cache space allocation in a large range of channel reliability levels.
Unequal Allocation (UA) Policy: Set $\vec{\mu} = (0, 0, \cdots, 1)$ and

$$\vec{b}(x) = \begin{cases} \vec{b}^\circ(x), & \text{if } p > p_{\text{th}}, \\ \vec{b}_{EA}, & \text{otherwise}, \end{cases}$$

where $\vec{b}_{EA} = (1/M, \cdots, 1/M)$ is the equal allocation vector, p_{th} is the unique solution to $L(M, \vec{b}_{EA}) = L(M, \vec{b}^\circ(x))$. Let $\mathbb{P}_{\text{miss}}^{\text{UA}}$ denote the miss probability of the UA policy.

Note that as the total cache space goes to infinity, the miss probabilities under the EA and the UA policies will all converge to the probability that no cache is accessible when the request arrives, i.e., to $(1-p)^M$. The following theorem characterizes how much faster the UA policy converges to this limit compared to the EA policy.

Theorem 5. *Define $\rho \triangleq \lim_{x\to\infty} \frac{\mathbb{P}_{miss}^{UA} - (1-p)^M}{\mathbb{P}_{miss}^{EA} - (1-p)^M}$, which measures how much faster the unequal cache space allocation policy converges to the limit $(1-p)^M$ compared to the equal cache space allocation policy. Then, we have*

$$\rho = \min\left\{1, \frac{L\left(M, \lim_{x\to\infty} \vec{b}^\circ(x)\right)}{L(M, \vec{b}_{EA})}\right\},$$

which is strictly less than 1 if $p > p_{th}$.

The proof is provided in Section VIII-G. Setting $\alpha = 1.4$,

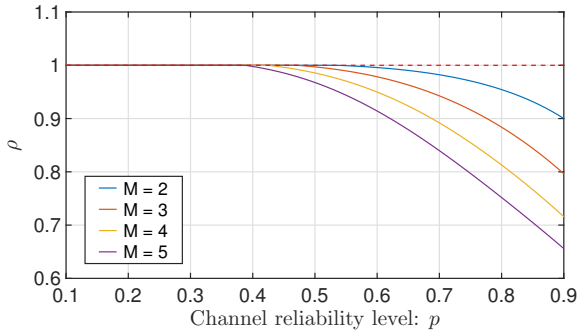

Fig. 2. Benefits of the UA policy over the EA policy.

we plot ρ as a function of p for different M's in Fig. 2. It can be observed that in a large range of channel reliabilities, e.g., $p \in (0.6, 0.9)$, ρ is much smaller than 1, which indicates that the UA policy gains considerable improvements over equal cache space allocation.

V. Generalization to Heterogenous Channels

In this section, we first generalize our model for heterogenous channel reliability levels and derive the asymptotic miss probabilities. Then, we leverage the optimal static caching policy to guide the cache space allocation for LRU caches and propose a generalized UA policy.

A. Miss probability

Consider the model described in Section II. Assume that the channels have heterogenous reliability levels. Specifically, let p_m denote the probability that cache C_m is accessible, $1 \leq m \leq M$. To obtain compact descriptions, we set $p_0 = 0$ and $I_{k,0}^{(m)} = 0$ for $1 \leq m \leq M$, $1 \leq k \leq \binom{M}{m}$. Recall that $I_{k,i}^{(m)}$'s are defined in Section IV-A as the indices of the caches in the set $\mathcal{S}_k^{(m)}$. Assume without loss of generality that ${b_m}^{\alpha}/p_m$ is non-increasing with respect to m, $1 \leq m \leq M$. We derive the asymptotic miss probability in the following theorem.

Theorem 6. *For M caches with heterogenous channel reliability levels, as the total cache size $x \to \infty$, we have, almost surely for all H,*

$$\mathbb{P}[\textit{Miss for } R_0|H] - \sum_{m=1}^{M} \sum_{k=1}^{\binom{M}{m}} W_k^{(m)} \prod_{i=1}^{m} \left(1 - p_{I_{k,i}^{(m)}}\right) \sim \left(\sum_{m=1}^{M} L\left(m, \vec{b}\right) \mu_m\right) \left(\sum_{m=1}^{M} m\mu_m\right)^{\alpha-1} Q(x),$$

and

$$\mathbb{P}[\textit{Miss for } R_0] - \sum_{m=1}^{M} \frac{\mu_m}{\binom{M}{m}} \sum_{k=1}^{\binom{M}{m}} \prod_{i=1}^{m} \left(1 - p_{I_{k,i}^{(m)}}\right) \sim \left(\sum_{m=1}^{M} L\left(m, \vec{b}\right) \mu_m\right) \left(\sum_{m=1}^{M} m\mu_m\right)^{\alpha-1} Q(x),$$

where $Q(x)$ is defined in (1) and

$$L\left(m, \vec{b}\right) = \frac{1}{\binom{M}{m}} \sum_{k=1}^{\binom{M}{m}} \sum_{(j,l):\mathcal{S}_j^{(l)} \subseteq \mathcal{S}_k^{(m)}} \left(\frac{\prod_{i=1}^{m} \left(1 - p_{I_{k,i}^{(m)}}\right)}{\prod_{i=1}^{l} \left(1 - p_{I_{j,i}^{(l)}}\right)} \cdot \prod_{i=1}^{l} p_{I_{j,i}^{(l)}} \left(\sum_{i=1}^{l} \left(M b_{I_{j,i}^{(l)}}\right)^{\alpha} \prod_{k=0}^{i-1} \left(1 - p_{I_{j,k}^{(i)}}\right) \right)^{1/\alpha - 1} \right).$$

The proof is presented in Section VIII-H. For a given total memory size x, a hashing mechanism $\vec{\mu}$ and a space allocation strategy $\vec{b}$, the asymptotic miss probability can be explicitly approximated by the function

$$P(x; \vec{\mu}, \vec{b}) \triangleq \left(\sum_{m=1}^{M} L\left(m, \vec{b}\right) \mu_m\right) \left(\sum_{m=1}^{M} m\mu_m\right)^{\alpha-1} Q(x) + \sum_{m=1}^{M} \frac{\mu_m}{\binom{M}{m}} \sum_{k=1}^{\binom{M}{m}} \prod_{i=1}^{m} \left(1 - p_{I_{k,i}^{(m)}}\right). \quad (6)$$

As expected, (6) will degenerate to (2) when all channels have the same reliability level. Moreover, using a similar approach that proves Theorem 2, it is easy to verify that for large x, the optimal hashing mechanism is to hash the data to all caches.

B. Cache space allocation

In this section, we will investigate the cache space allocation for general channel reliability levels under the assumption that the data items are hashed to all caches. Similar to the analysis in Section IV-C, we leverage the optimal static caching policy to guide our design for LRU caches, and use the same notations defined therein.

Lemma 3 (An Optimal Static Policy for General Channels)**.** *Assume that the channel reliability p_m is non-increasing with respect to m. If the popularity of each content were known a priori, the following static policy minimizes the miss probability for M caches with a total memory space x:*
Cache space allocation: Set $x_1^\circ \geq x_2^\circ \geq \cdots \geq x_M^\circ$, satisfying

$$\frac{p_j \Pi_{i=0}^{j-1}(1-p_i)}{(x_j^\circ)^\alpha} \geq \frac{p_k \Pi_{i=0}^{k-1}(1-p_i)}{(x_k^\circ + 1)^\alpha} \quad \textit{for } \forall x_j^\circ \geq 1, x_k^\circ \geq 1,$$

$$\sum_{m=1}^{M} x_m^\circ = \lfloor x \rfloor;$$

Data placement: Store data items $\{d_i : 1 \leq i \leq x_m^\circ\}$ in cache C_m, if $x_m^\circ \geq 1$.

The proof of Lemma 3 is presented in Section VIII-I As the cache space goes to infinity, the optimal static allocation can be explicitly calculated as

$$\lim_{x \to \infty} b_m^\circ(x) = \frac{p_m \Pi_{i=0}^{m-1}(1-p_i)}{\sum_{k=1}^{M} p_k \Pi_{i=0}^{k-1}(1-p_i)}. \quad (7)$$

Key Insights: For channels with heterogenous reliability levels, the optimal static cache space allocation (7) is even more skewed than the one for identical channels (5), because more benefits can be obtained by utilizing the most reliable channels. Moreover, the key insights revealed for homogenous channels in Section IV-C also hold for heterogenous channels. Using a similar approach that proves Theorem 4, we can verify that the optimal allocation is also unequal for LRU caching over heterogenous channels.

We will use the optimal static solution as the allocation method for LRU caches.

Generalized Unequal Allocation (UA-G) Policy: Set $\vec{\mu} = (0, 0, \cdots, 1)$ and

$$\vec{b}(x) = \begin{cases} \vec{b}^\circ(x), & \text{if } L(M, \vec{b}^\circ(x)) < L(M, \vec{b}_{EA}), \\ \vec{b}_{EA}, & \text{otherwise,} \end{cases}$$

where $\vec{b}^\circ(x)$ can be obtained from Lemma 3, $\vec{b}_{EA} = (1/M, 1/M, \cdots, 1/M)$ is the equal allocation vector, and the function $L(M, \vec{b})$ is defined in Theorem 6.

The UA-G policy is a generalized UA policy designed for general channel reliability levels. It is verified in Experiment 5 that, the UA-G policy outperforms the UA policy when the channel reliability levels are heterogenous. In addition, a

performance guarantee can be derived for the UA-G policy by simply replacing the function $L(M,\vec{b})$ in Theorem 5 by the generalized $L(M,\vec{b})$ function defined in Theorem 6. Due to the limited space, we omit this theorem.

VI. Experiments

To validate our theoretical analysis, we conduct 5 experiments. In Experiment 1, we simulate 5 caches with general $\vec{b}$ and $\vec{\mu}$, which validates Theorem 1. In Experiments 2 and 3, we compare the equal cache space allocation with resource pooling and unequal cache space allocation, respectively. In Experiment 4, we evaluate the proposed policies using real data traces. In Experiment 5, we compare the UA-G policy with the UA policy when the channels have heterogenous reliability levels. Experiment results successfully validate the counterintuitive insights revealed by theoretical analyses.

Experiment 1. In this experiment, we validate Theorem 1 by simulating 5 caches over unreliable channels. Let $\vec{b} = (0.3, 0.2, 0.2, 0.15, 0.15)$, $\vec{\mu} = (0.1, 0.15, 0.2, 0.25, 0.3)$. Set $q_i = c/i^{1.8}$ for $1 \le i \le 10^7$, where $c = 1/\sum_{i=1}^{10^7} i^{-1.8} \approx 0.5313$. In Fig 3, we plot the miss probabilities for $p = 0.8, 0.85, 0.9$, respectively. The theoretical results approximated by Theorem 1 match very well with the empirical ones obtained by simulations, even when the total cache space is relatively small.

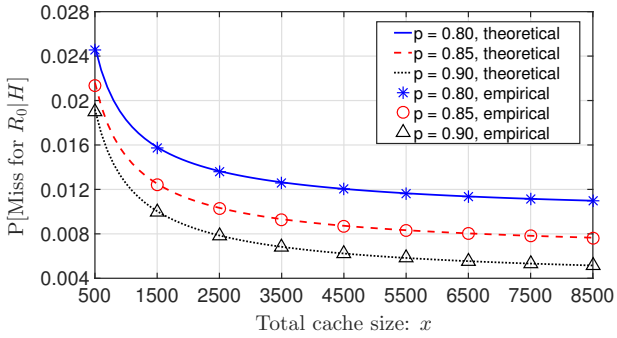

Fig. 3. Multiple unreliable caches accessed by hashing.

Experiment 2. In this experiment, we compare the miss probabilities achieved by the RP policy with that achieved by the EA policy. Set $p = 0.9$ and $q_i = c/i^{1.8}$ for $1 \le i \le 10^7$, where $c = 1/\sum_{i=1}^{10^7} i^{-1.8} \approx 0.5313$. In Fig. 4, we plot the miss probabilities for $M = 1, 2, 3, 4$, respectively. It can be observed that resource separation ($M \ge 2$) achieves much better miss probabilities than resource pooling ($M = 1$), which validates our statements in Theorem 2. Moreover, as what we comment in Theorem 3, allocating the total cache space to more caches may not guarantee a better miss probability (e.g., allocating the total cache space to 3 caches achieves lower miss probabilities than allocating to 4 caches when $x \in (100, 900)$). In fact, since the theoretical approximations for miss probabilities are sufficiently accurate even when the cache space is relatively small (e.g., $x = 200$), the optimal number of caches can be theoretically calculated by minimizing $L(M, (1/M, \cdots, 1/M))$ over M.

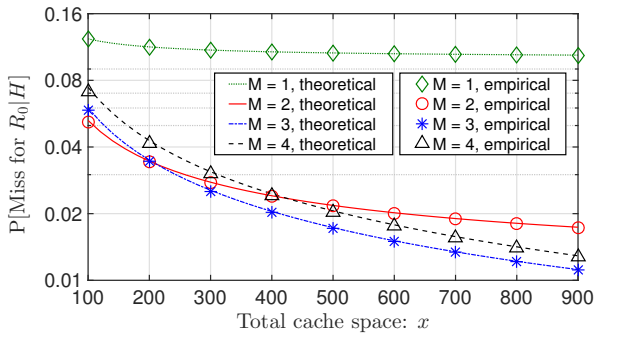

Fig. 4. Resource pooling v.s. equal cache space allocation.

Experiment 3. In this experiment, we compare the EA policy with the UA policy. Let $M = 5, q_i = c/i^2$ for $1 \le i \le 10^7$, where $c = 1/\sum_{i=1}^{10^7} i^{-2} \approx 0.6079$. First, setting the total cache space $x = 500$ and applying Theorem 5, we compute ρ under different channel reliability levels and plot the results in Fig.5[left]. Then, setting $p = 0.7$, we plot the miss probabilities achieved by the EA policy and the UA policy in Fig. 5[right]. All empirical results match well with theoretical ones. It can be observed from Fig. 5[left] that when p is greater than the threshold p_{th} (≈ 0.4), ρ will be strictly less than one (i.e., the UA policy outperforms the EA policy). Furthermore, by setting $p = 0.7$, it is shown in Fig. 5[right] that the miss probabilities achieved by the UA policy (unequal cache space allocation) can be significantly smaller than that achieved by the EA policy (equal cache space allocation). For $p > 0.7$, an even larger improvement is expected.

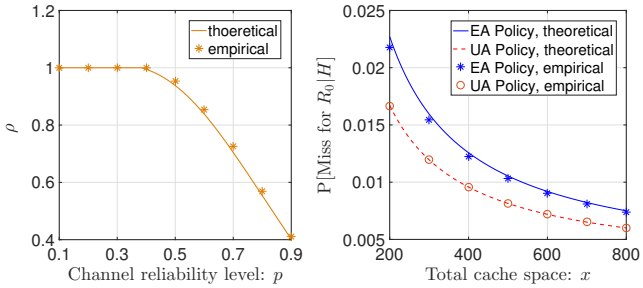

Fig. 5. Equal v.s. unequal cache space allocation.

Experiment 4. In this experiment, we compare the RP, EA and UA policies using a data trace collected on a content delivery network. The trace is also used for evaluation and labeled as cdn1 in [31], [32]. Our objective is to check whether our designs perform well under popularity distributions obtained

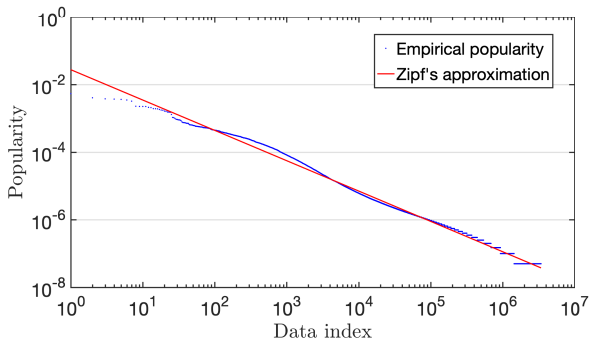

Fig. 6. Popularity of trace data.

from real-world traces. In this experiment, we compare the miss ratios of 20 million requests [1] under different policies. We plot the empirical popularities in Fig. 6, as well as the Zipf's approximation (i.e., $0.0273/i^{0.897}$, $1 \leq i \leq 3417123$). The UA policy is designed using Equation (5) with $\alpha = 0.897$. Notably, $\alpha = 0.897$ is less than 1 and therefore beyond the scope of this paper (see [23] for LRU caching with $\alpha < 1$). However, the insights revealed by our theoretical analyses still hold according to the following experiment results. Set $M = 2$. For a fixed

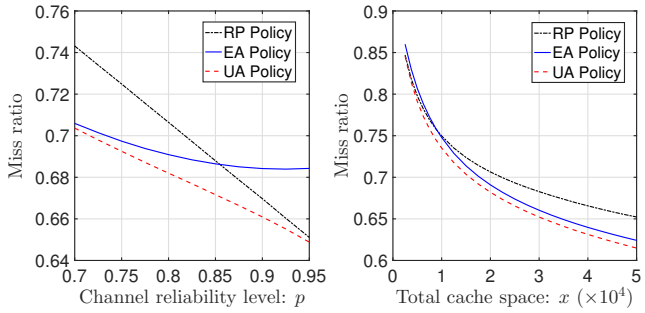

Fig. 7. Performance evaluation based on real-world data traces.

$x = 20000$, we first compare the miss ratios of the RP, EA and UA policies under different channel reliability levels, and plot the results in Fig. 7[left]. It can be observed that the UA policy always achieves the best miss ratios. Moreover, when p is relatively large (respectively small), the EA (respectively RP) policy achieves much worse performance, which validates the insights revealed by Theorem 5. Next, for a fixed $p = 0.8$, we compare the proposed policies under different cache spaces. The results are plotted in Fig. 7[right]. It can be observed that the UA policy outperforms the RP and EA policies in the whole range of x.

Experiment 5. In this experiment, we consider heterogenous channel reliability levels and compare the miss probabilities achieved by the UA-G policy and the UA policy. Let $x = 400, M = 5, q_i = c/i^2$ for $1 \leq i \leq 10^7$, where $c = 1/\sum_{i=1}^{10^7} i^{-2} \approx 0.6079$. Let the channel reliability levels be $p_m = -a(m-3)+0.7, 1 \leq m \leq 5, a \in (0, 0.15)$. Note that the average reliability level $\sum_{m=1}^{5} p_m/5 = 0.7$ is a constant and the parameter a represents the skewness of the reliability levels. When $a = 0$, the channels are homogenous and all have the same reliablity level 0.7. When $a = 1.5$, we have $p_1 = 1$ and $p_5 = 0.4$, i.e., the skewness is maximized. We

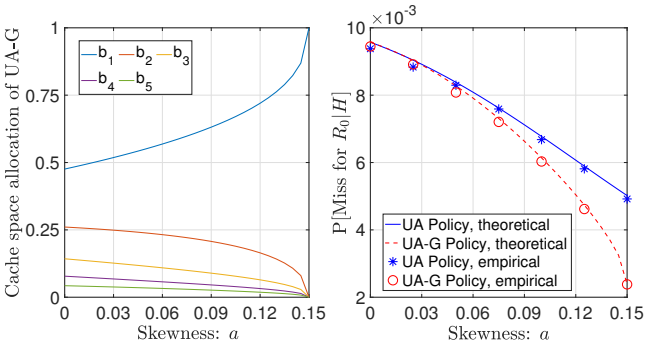

Fig. 8. Generalized unequal cache space allocation for heterogenous channels.

[1]Use the first 20 million requests of the trace with item sizes set to be 1.

apply the UA-G policy for different channel reliability levels and compare it to the UA policy that is computed by setting $p = 0.7$. In Fig. 8[left], we plot the cache space allocation of the UA-G policy. Note that when $a = 0$, the allocation of the UA-G policy is the same as that of the UA policy, since the channels have identical reliability levels. As the skewness a increases, more cache space will be allocated to the most reliable cache. When $a = 0.15$ (i.e., the first channel is always reliable), all cache space will be allocated to the first cache. In Fig. 8[right], we compare the miss probability achieved by the UA-G and the UA policies. It can be observed that, the UA-G policy achieves better miss probabilities than the UA policy when $a > 0$ (i.e., the channels are heterogenous). And the gain is increasing with respect to the reliability skewness a. Moreover, the empirical results match well with the theoretical ones, which validates Theorem 6.

VII. Conclusion

In this work, we studied the distributed LRU caching over unreliable channels by explicitly approximating the miss probability and discovering counterintuitive insights in optimizing the cache space allocation and the data placement. Our investigation revealed two counterintuitive insights that are in stark contrast with the principles of distributed LRU caching under reliable conditions: (i) that resources-pooling is no longer optimal in the presence of channel unreliabilities, and (ii) that, even under symmetric unreliabilities, it is necessary to allocate unequal cache spaces to otherwise identical distributed caches. Our analysis framework also allowed us to develop an explicit unequal allocation policy which outperforms the equal allocation in a large range of channel reliability levels. These insights and designs are expected to help with the development and implementation of efficient distributed caching solutions under unreliable conditions.

VIII. Proofs

A. Proof of Lemma 1

Proof. If $p = 1$, i.e., the caches are always accessible, storing the same data item in multiple caches will waste the cache space and bring no additional benefits. Therefore, the optimal hashing vector is $\vec{\mu}^* = (1, 0, \cdots, 0)$ i.e., hashing each data item to only one cache. Then, applying Theorem 2 of [26], we have $\mathbb{P}[\text{Miss for } R_0|H] \sim Q(\bar{x})$, where

$$\bar{x} = \left(\sum_{m=1}^{M} {b_m}^{1-\alpha} M^{-\alpha} \right)^{-1/(\alpha-1)} x.$$

Moreover, applying the Hoeffding's inequality, we have $\sum_{m=1}^{M} {b_m}^{1-\alpha} M^{-\alpha} \geq 1$, where the equality holds if and only if $b_m = 1/M$ for $1 \leq m \leq M$. Since $Q(x)$ is decreasing with x, we finish the proof. □

B. Proof of Theorem 1

To prove Theorem 1, we establish the following lemmas.

Lemma 4. *Consider M caches organized in the system described in Section II. Let the data items be hashed to all*

caches. Assume $\mathcal{J}_0 = \{C_1, \cdots, C_m\}$, $1 \le m \le M$ (i.e., the first m caches are accessible at τ_0). As the total cache space $x \to \infty$, we have

$$\mathbb{P}[\textit{Miss for } R_0 | \mathcal{J}_0 = \{C_1, \ldots, C_m\}, H] \sim Q(\bar{x})$$

where $\bar{x} = \left(\sum_{k=1}^{m} (1-p)^{k-1} {b_k}^{\alpha}\right)^{1/\alpha} x$.

Proof. For $1 \le k \le M$, let $\tau_{-\sigma_k}$ denote the last time before τ_0 when the data item R_0 was requested and the cache C_k was accessible then. Define

$$T_k(n) = \sum_{i \ge 1} \mathbf{1}\left(\cup_{j=1}^{n} \{\{R_{-j} = d_i\} \cap \{C_k \in \mathcal{I}_{-j}\}\}\right),$$

where $\mathbf{1}(\mathcal{E})$ is the indicator function having the value 1 when the event $\mathcal{E}$ happens and the value 0 otherwise. $T_k(n)$ represents the number of distinct data requests that successfully access cache C_k between time τ_{-n} and time τ_{-1}. We also define the inverse of $T_k(n)$ as $T_k^{\leftarrow}(x) = \min\{n : T_k(n) \ge x\}$.

It can be verified that

$$\begin{aligned} &\{\text{Miss for } R_0 | \mathcal{J}_0 = \{C_1, \cdots, C_m\}, H\} \\ &\Leftrightarrow \cap_{k=1}^{m} \{\sigma_k > T_k^{\leftarrow}(b_k x)\}. \end{aligned} \tag{8}$$

Note that, according to the definition of $\tau_{-\sigma_k}$ and the LRU policy, R_0 is cached in C_k right after $\tau_{-\sigma_k}$. On the one hand, if a miss happens at τ_0, i.e., R_0 is not stored in any caches at τ_0, then there must be sufficient requests arriving between $\tau_{-\tau_k}$ and τ_0 such that R_0 was evicted from the cache. On the other hand, if the size of all distinct data requests arriving between $\tau_{-\tau_k}$ and τ_0 exceeds the cache size, then a miss will happen at time τ_0.

The rest of the proof is consisted of two steps. In Step 1, we will estimate $\mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n_k\}]$, where n_k's are given constants. In Step 2, we show that $\mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > T_k^{\leftarrow}(b_k x)\}]$ can be approximated by $\mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > \bar{T}^{\leftarrow}(b_k x)\}]$, where

$$\bar{T}^{\leftarrow}(x) \approx \Gamma(1 - 1/\alpha)^{-\alpha} c^{-1} p^{-1} x^{\alpha}.$$

Step 1: For constants $n_1 > n_2 > \cdots > n_m > n_{m+1} = 0$ and $\forall d_i \in \mathcal{D}$, we have

$$\begin{aligned} &\mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n_k\} | R_0 = d_i] \\ &= \mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n_m\} | R_0 = d_i] \\ &\qquad \cdot \mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n_k\} | \cap_{k=1}^{m} \{\sigma_k > n_m\}, R_0 = d_i] \\ &= \mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n_m\} | R_0 = d_i] \\ &\qquad \cdot \mathbb{P}[\cap_{k=1}^{m-1} \{\sigma_k > n_k - n_m\} | R_0 = d_i] \\ &= \mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n_m\} | R_0 = d_i] \\ &\qquad \cdot \Pi_{j=1}^{m-1} \mathbb{P}[\cap_{k=1}^{j} \{\sigma_k > n_j - n_{j+1}\} | R_0 = d_i] \\ &= \Pi_{j=1}^{m} \mathbb{P}[\cap_{k=1}^{j} \{\sigma_k > n_j - n_{j+1}\} | R_0 = d_i], \end{aligned} \tag{9}$$

where the second equality holds since the requests are assumed to arrive according to independent Poisson processes. For $\forall n \ge 1$, let $Y_i(n) = \sum_{j=1}^{n} \mathbf{1}(\{R_{-j} = d_i\})$, $1 \le i \le N$. $Y_i(n)$ represents the number of requests that fetch data d_i during τ_{-n} and τ_{-1}, and follows a binomial distribution. We have,

$$\begin{aligned} &\mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n\} | R_0 = d_i] \\ &= \sum_{j=0}^{n} \mathbb{P}[Y_i(n) = j] \mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n\} | Y_i(n) = j] \\ &= \sum_{j=0}^{n} \mathbb{P}[Y_i(n) = j] (1-p)^{mj} \\ &= \mathbb{E}\left[\exp(m \log(1-p) Y_i(n))\right] \\ &= \mathcal{M}_{Y_i(n)}(m \log(1-p)), \end{aligned}$$

where $\mathcal{M}_{Y_i(n)}(t) \triangleq \mathbb{E}[\exp(t Y_i(n))]$ is the moment generating function of $Y_i(n)$. Recalling that $Y_i(n)$ follows a binomial distribution $B(n, q_i)$, we have $\mathcal{M}_{Y_i(n)}(t) = (1 - q_i + q_i e^t)^n$, which implies

$$\mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n\} | R_0 = d_i] = (1 - q_i + q_i (1-p)^m)^n. \tag{10}$$

By combining (9) and (10), we have

$$\begin{aligned} &\mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n_k\}] = \sum_{i=1}^{\infty} q_i \mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n_k\} | R_0 = d_i] \\ &= \sum_{i=1}^{\infty} q_i \Pi_{k=1}^{m} \left(1 - q_i + q_i (1-p)^k\right)^{n_k - n_{k+1}}. \end{aligned}$$

Assume that there exist $c_1 > c_2 > \cdots > c_{m-1} > c_m = 1$ such that

$$n_k = c_k n_m, \text{ for } 1 \le k \le m. \tag{11}$$

Define $n_{m+1} = c_{m+1} = 0$. We will show that, as $n_m \to \infty$,

$$\begin{aligned} &\mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n_k\}] \\ &\sim \sum_{i=1}^{\infty} q_i \exp\left(-q_i \sum_{k=1}^{m} (1 - (1-p)^k)(n_k - n_{k+1})\right) \\ &\sim \frac{c^{1/\alpha} \Gamma(2 - 1/\alpha)}{\alpha - 1} \left(p \sum_{k=1}^{m} (1-p)^{k-1} n_k\right)^{-1+1/\alpha}. \end{aligned} \tag{12}$$

On the one hand, due to the fact that $1 - x < e^{-x}$ for $x \in (0, 1)$, we have

$$\begin{aligned} &\mathbb{P}[\cap_{k=1}^{m} \{\sigma_k > n_k\}] \\ &= \sum_{i=1}^{\infty} q_i \Pi_{k=1}^{m} \left(1 - q_i + q_i (1-p)^k\right)^{n_k - n_{k+1}} \\ &< \sum_{i=1}^{\infty} q_i \exp\left(-q_i \sum_{k=1}^{m} (1 - (1-p)^k)(n_k - n_{k+1})\right) \\ &= \sum_{i=1}^{\infty} q_i \exp\left(-q_i p \sum_{k=1}^{m} (1-p)^{k-1} n_k\right). \end{aligned} \tag{13}$$

According to Theorem 3.6 in [10], we have, as $n \to \infty$

$$\begin{aligned} \sum_{i=1}^{\infty} q_i (1 - q_i)^n &\sim \sum_{i=1}^{\infty} q_i \exp(-q_i n) \\ &\sim \frac{\Gamma(2 - 1/\alpha)}{\alpha - 1} c^{1/\alpha} n^{-1+1/\alpha}. \end{aligned} \tag{14}$$

Combining (13) and (14) implies

$$\mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > n_k\}] \lesssim \frac{c^{1/\alpha}\Gamma(2-1/\alpha)}{\alpha-1}\left(p\sum_{k=1}^{m}(1-p)^{k-1}n_k\right)^{-1+1/\alpha}, \quad (15)$$

as $n_m \to \infty$.

On the other hand, for any $\delta \in (0,1)$, there exists $x_\delta > 0$ such that $1-x > e^{-(1+\delta)x}$ for $0 < x < x_\delta$. Selecting i_δ such that $q_{i_\delta} < x_\delta$, we have, as $n_m \to \infty$,

$$\begin{aligned}
&\mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > n_k\}] \\
&\geq \sum_{i=i_\delta}^{\infty} q_i \Pi_{k=1}^{m}\left(1-q_i+q_i(1-p)^k\right)^{n_k-n_{k+1}} \\
&\geq \sum_{i=i_\delta}^{\infty} q_i \exp\left(-q_i\sum_{k=1}^{m}(1-(1-p)^k)(n_k-n_{k+1})(1+\delta)\right) \\
&= \sum_{i=1}^{\infty} q_i \exp\left(-q_i\sum_{k=1}^{m}(1-(1-p)^k)(n_k-n_{k+1})(1+\delta)\right) \\
&\quad - \sum_{i=1}^{i_\delta-1} q_i \exp\left(-q_i\sum_{k=1}^{m}(1-(1-p)^k)(n_k-n_{k+1})(1+\delta)\right) \\
&\sim \frac{c^{1/\alpha}\Gamma(2-1/\alpha)}{\alpha-1}\left(p\sum_{k=1}^{m}(1-p)^{k-1}n_k(1+\delta)\right)^{-1+1/\alpha}.
\end{aligned}$$

Therefore, we have, as $n_m \to \infty$ and $\delta \to 0$,

$$\mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > n_k\}] \gtrsim \frac{c^{1/\alpha}\Gamma(2-1/\alpha)}{\alpha-1}\left(p\sum_{k=1}^{m}(1-p)^{k-1}n_k\right)^{-1+1/\alpha}. \quad (16)$$

Combining (15) and (16) implies (12). Up to now, we have finished the first step of the proof.

Step 2: Next, we will show that

$$\mathbb{P}[\text{Miss for } R_0|\mathcal{J}_0 = \mathcal{C}, H] \sim \frac{c^{1/\alpha}\Gamma(2-1/\alpha)}{\alpha-1}\left(p\sum_{k=1}^{m}(1-p)^{k-1}\bar{T}^{\leftarrow}(b_k x)\right)^{-1+1/\alpha},$$

where

$$\bar{T}(n) \triangleq \sum_{i=1}^{\infty}\left(1-(1-q_i+q_i(1-p))^n\right)$$
$$\bar{T}^{\leftarrow}(x) \triangleq \min\{n : \bar{T}(n) \geq x\}.$$

We have, as $n \to \infty$

$$\bar{T}(n) \sim \sum_{i=1}^{\infty}(1-\exp(-q_i p n)) \sim \Gamma\left(1-\frac{1}{\alpha}\right)(cpn)^{1/\alpha},$$

and as $x \to \infty$

$$\bar{T}^{\leftarrow}(x) \sim \Gamma(1-1/\alpha)^{-\alpha}c^{-1}p^{-1}x^{\alpha}. \quad (17)$$

Applying Lemma 7.1 in [10], we can prove that for $\forall \epsilon > 0$,

$$\mathbb{P}\left[T_k^{\leftarrow}(x) < \bar{T}^{\leftarrow}\left(\frac{x}{1+\epsilon}\right)\right] \leq \exp\left(-\frac{\epsilon^2 x}{4(1+\epsilon)}\right), \quad (18)$$
$$\mathbb{P}\left[T_k^{\leftarrow}(x) > \bar{T}^{\leftarrow}\left(\frac{x}{1-\epsilon}\right)\right] \leq \exp\left(-\frac{\epsilon^2 x}{4(1-\epsilon)}\right).$$

Combining (8), (12), (17) and (18), we have for $\forall \epsilon > 0$, there exists an x_0 such that, for $x \geq x_0$

$$\begin{aligned}
&\mathbb{P}[\text{Miss for } R_0|\mathcal{I}_0 = \mathcal{J}_0 = \{C_1,\cdots,C_m\}, H] \\
&\leq \mathbb{P}[\sigma_1 > \bar{T}^{\leftarrow}(b_1 x/(1+\epsilon)),\cdots,\sigma_m > \bar{T}^{\leftarrow}(b_m x/(1+\epsilon))] \\
&\quad + \sum_{k=1}^{m}\mathbb{P}[T_k^{\leftarrow}(b_k x) < \bar{T}^{\leftarrow}(b_k x/(1+\epsilon))] \\
&\leq Q\left(\left(\sum_{k=1}^{m}(1-p)^{k-1}{b_k}^{\alpha}\right)^{1/\alpha}\frac{x}{1+\epsilon}\right) \\
&\quad + \sum_{k=1}^{m}\exp\left(-\frac{\epsilon^2 b_k x}{4(1+\epsilon)}\right) \\
&= Q\left(\bar{x}/(1+\epsilon)\right) + o(Q(\bar{x})), \quad (19)
\end{aligned}$$

where $\bar{x} = \left(\sum_{k=1}^{m}(1-p)^{k-1}{b_k}^{\alpha}\right)^{1/\alpha}x$. Similarly, we can prove for large x

$$\begin{aligned}
&\mathbb{P}[\text{Miss for } R_0|\mathcal{I}_0 = \mathcal{J}_0 = \{C_1,\cdots,C_m\}, H] \\
&\geq Q\left(\bar{x}/(1-\epsilon)\right) - o(Q(\bar{x})). \quad (20)
\end{aligned}$$

Combining (19) and (20) then letting $x \to \infty$ finish the proof. □

Lemma 5. *Data items that are hashed to the cache set $\mathcal{S}_k^{(m)}$ asymptotically follow a Zipf's distribution $c_k^{(m)}/i^{\alpha}, i \geq 1$ almost surely for all H, where*

$$c_k^{(m)} = \frac{c{\mu_m}^{\alpha}}{W_k^{(m)}\binom{M}{m}^{\alpha}}.$$

Proof. Let $X_{i,k}^{(m)} = 1$ indicate that $H(d_i) = \mathcal{S}_k^{(m)}$. Otherwise, $X_{i,k}^{(m)} = 0$. Define $I_{n,k}^{(m)} = \sum_{i=1}^{n} X_{i,k}^{(m)}$. Applying the Bernstein's inequality (Theorem 2.8 in [33]), we can prove that for $\forall 0 < \epsilon < 1$, there exists n_1 such that for $\forall n > n_1$,

$$\begin{aligned}
&\mathbb{P}\left[\left|\frac{I_{n,k}^{(m)}}{n\mu_m/\binom{M}{m}} - 1\right| > \epsilon\right] \leq 2\exp\left(-\frac{n^2{\mu_m}^2\epsilon^2/\binom{M}{m}^2}{2n\mu_m/\binom{M}{m} + 2\epsilon/3}\right) \\
&\leq 2\exp\left(-\frac{\mu_m n}{3\binom{M}{m}}\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{P}\left[\bigcap_{i\geq n}\left\{\left|\frac{I_{i,k}^{(m)}}{n\mu_m/\binom{M}{m}} - 1\right| \leq \epsilon\right\}\right] \\
&\geq 1 - \sum_{i\geq n}\mathbb{P}\left[\left|\frac{I_{i,k}^{(m)}}{n\mu_m/\binom{M}{m}} - 1\right| > \epsilon\right] \\
&\geq 1 - 2\sum_{i\geq n}\exp\left(-\frac{\mu_m i}{3\binom{M}{m}}\right) \\
&\geq 1 - 2\int_{n-1}^{\infty}\exp\left(-\frac{\mu_m t}{3\binom{M}{m}}\right)dt.
\end{aligned}$$

There exist a large integer n_2 and a constant c_1 such that $2\int_{n-1}^{\infty}\exp\left(-\mu_m t/\left(3\binom{M}{m}\right)\right)dt < c_1/n^2$ for all $n > n_2$, which implies

$$\mathbb{P}\left[\bigcap_{i\geq n}\left\{\left|\frac{I_{i,k}^{(m)}}{n\mu_m/\binom{M}{m}}-1\right|\leq\epsilon\right\}\right]\geq 1-\frac{c_1}{n^2}.$$

Since $\sum_{n=1}^{\infty} c_1/n^2<\infty$, by the Borel-Cantelli lemma, for any ϵ, there always exists an integer $n_H=\max\{n_1,n_2\}$ such that for $\forall n>n_H$

$$\bigcap_{i\geq n}\left\{\left|\frac{I_{i,k}^{(m)}}{i\mu_m/\binom{M}{m}}-1\right|\leq\epsilon\right\}$$

holds almost surely for any H.

Let $q_{i,k}^{(m)}$ denote the popularity of the data item with the index $I_{i,k}^{(m)}$. We have almost surely for all H, $q_{i,k}^{(m)}\sim c\Big/\left(i\binom{M}{m}\Big/\mu_m\right)^{\alpha}$. Then normalizing $q_{i,k}^{(m)}$ by $\mathbb{P}[\mathcal{I}_0=\mathcal{S}_k^{(m)}]=W_k^{(m)}$ finishes the proof. □

Lemma 6 (Theorem 4.1 of [10])**.** *Consider K flows of independent data requests sharing a LRU cache. For each flow k, $1\leq k\leq K$, the data popularity follows a Zipf's distribution c_k/i^{α}, $\alpha>1$, $i\geq 1$ asymptotically. Assume the size of each data item is 1. Let ν_k denote the probability that a request is from flow k, then as the cache space x goes to infinity, we have*

$$\mathbb{P}[\textit{Miss for } R_0|R_0 \textit{ is from flow } k]\sim Q\left(\bar{x}_k\right),$$

where

$$\bar{x}_k=\frac{(c_k\nu_k)^{1/\alpha}x}{\sum_{i=1}^{K}(c_i\nu_i)^{1/\alpha}}. \tag{21}$$

With the established lemmas, now we are ready to prove Theorem 1.

Proof of Theorem 1. Under the model described in Section II, we have,

$$\begin{aligned}
&\mathbb{P}[\text{Miss for } R_0|H]\\
&=\sum_{m=1}^{M}\sum_{k=1}^{\binom{M}{m}}\mathbb{P}[\mathcal{I}_0=\mathcal{S}_k^{(m)}]\mathbb{P}[\text{Miss for } R_0|\mathcal{I}_0=\mathcal{S}_k^{(m)},H]\\
&=\sum_{m=1}^{M}\sum_{k=1}^{\binom{M}{m}}\mathbb{P}[\mathcal{I}_0=\mathcal{S}_k^{(m)}]\Big((1-p)^m\\
&\quad+\sum_{(i,j):\mathcal{S}_i^{(j)}\subseteq\mathcal{S}_k^{(m)}}\Big(\mathbb{P}[\mathcal{J}_0=\mathcal{S}_i^{(j)}|\mathcal{I}_0=\mathcal{S}_k^{(m)},H]\\
&\qquad\cdot\mathbb{P}[\text{Miss for } R_0|\mathcal{J}_0=\mathcal{S}_i^{(j)},\mathcal{I}_0=\mathcal{S}_k^{(m)},H]\Big)\Big).
\end{aligned}\tag{22}$$

We can view the data requests that are hashed to the same set of caches as a data flow. Then, each cache is shared by multiple data flows. For example, consider a system with 2 caches (C_1 and C_2). There are three possible hashing results, and therefore three data flows, i.e., data requests that are hashed to only C_1, only C_2, and both C_1 and C_2. In this example, each cache is shared by two data flows.

Lemma 5 implies that the popularity of each data flow is asymptotically a Zipf's distribution. Then, applying Lemma 6, we know that for multiple flows sharing a LRU cache with a total memory space x, it is equivalent (in terms of miss probabilities) to the case where each data flow (e.g., flow k) is served by a separate LRU cache with a virtual cache space ($\bar{x}_k$ defined in (21)). Specifically, consider the data flow with requests that are hashed to $\mathcal{S}_k^{(m)}$. Assume without loss of generality that $C_1\in\mathcal{S}_k^{(m)}$. Let $\mathcal{N}_k^{(m)}$ denote the set of all flows that are served by C_1. Then the fraction of the cache space of C_1 allocated to the flow is

$$\begin{aligned}
&\frac{\left(\frac{c{\mu_m}^{\alpha}}{W_k^{(m)}\binom{M}{m}^{\alpha}}\cdot W_k^{(m)}\right)^{1/\alpha}}{\sum_{(j,l)\in\mathcal{N}_k^{(m)}}\left(\frac{c{\mu_l}^{\alpha}}{W_j^{(l)}\binom{M}{l}^{\alpha}}\cdot W_j^{(l)}\right)^{1/\alpha}}=\frac{\mu_m/\binom{M}{m}}{\sum_{(j,l)\in\mathcal{N}_k^{(m)}}\mu_l/\binom{M}{l}}\\
&=\frac{\mu_m/\binom{M}{m}}{\sum_{l=1}^{M}\binom{M-1}{l-1}\mu_l/\binom{M}{l}}=\frac{M\mu_m}{\binom{M}{m}\sum_{l=1}^{M}l\cdot\mu_l}.
\end{aligned}\tag{23}$$

Note that this fraction is independent with C_1 and therefore should for all caches in $\mathcal{S}_k^{(m)}$. Combining Lemma 4, Lemma 5 and Lemma 6 yields

$$\begin{aligned}
&\mathbb{P}[\text{Miss for } R_0|\mathcal{J}_0=\mathcal{S}_i^{(j)},\mathcal{I}_0=\mathcal{S}_k^{(m)},H]\\
&\sim\frac{\Gamma(1-1/\alpha)^{\alpha}}{\alpha}\frac{c{\mu_m}^{\alpha}}{W_k^{(m)}\binom{M}{m}^{\alpha}\left(\bar{x}_i^{(j)}\right)^{\alpha-1}}
\end{aligned}\tag{24}$$

as $x\to\infty$, where

$$\bar{x}_i^{(j)}=\left(\sum_{l=1}^{j}(1-p)^{l-1}{b_{I_{i,l}^{(j)}}}^{\alpha}\right)^{1/\alpha}\frac{M\mu_m x}{\binom{M}{m}\sum_{l=1}^{M}l\,\mu_l}.$$

Reorganizing (24) yields that as $x\to\infty$,

$$\begin{aligned}
&\mathbb{P}[\text{Miss for } R_0|\mathcal{J}_0=\mathcal{S}_i^{(j)},\mathcal{I}_0=\mathcal{S}_k^{(m)},H]\\
&\sim\left(\sum_{i=1}^{M}i\mu_i\right)^{\alpha-1}\frac{\mu_m}{W_k^{(m)}\binom{M}{m}}\\
&\qquad\cdot\left(\sum_{l=1}^{j}(1-p)^{l-1}\left(Mb_{I_{i,l}^{(j)}}\right)^{\alpha}\right)^{1/\alpha-1}Q(x).
\end{aligned}\tag{25}$$

Plugging $\mathbb{P}[\mathcal{I}_0=\mathcal{S}_k^{(m)}|H]=W_k^{(m)}$, $\mathbb{P}[J_0=\mathcal{S}_i^{(j)}|I_0=\mathcal{S}_k^{(m)},H]=p^j(1-p)^{m-j}$ and (25) into (22) finishes the proof. □

C. Proof of Theorem 2

Proof. Recall that

$$\begin{aligned}
P\left(x;\vec{\mu},\vec{b}\right)&=\sum_{m=1}^{M}(1-p)^m\mu_m\\
&\quad+\left(\sum_{m=1}^{M}L(m,\vec{b})\mu_m\right)\left(\sum_{m=1}^{M}m\mu_m\right)^{\alpha-1}Q(x),
\end{aligned}$$

where

$$L(m,\vec{b})=\sum_{i=1}^{m}\binom{m}{i}p^i(1-p)^{m-i}\left(\frac{p}{1-(1-p)^i}\right)^{1-1/\alpha}.$$

Since $\lim_{x\to\infty} Q(x) = 0$, we have $\lim_{x\to\infty} P(x;\vec{\mu},\vec{b}) = \sum_{m=1}^{M}(1-p)^m\mu_m$, which implies $\lim_{x\to\infty}\mu_m^*(x) = 0$ for $1 \le m \le M-1$, and $\lim_{x\to\infty}\mu_M^*(x) = 1$.

□

D. Proof of Theorem 3

Proof. Assuming the data items are hashed to all caches, we have $\mathcal{I}_0 = \mathcal{C}$, which is no longer a random variable. Therefore, Theorem 1 as well as Corollary 1 still holds even when M scales with x as long as $\lim_{x\to\infty} x/M = \infty$. Corollary 1 yields $P\left(x;\vec{\mu},\vec{b}\right) = (1-p)^M + L(M,\vec{b})M^{\alpha-1}Q(x)$ where

$$L(M,\vec{b}) = \sum_{i=1}^{M}\binom{M}{i}p^i(1-p)^{M-i}\left(\frac{p}{1-(1-p)^i}\right)^{1-1/\alpha}.$$

Since $\lim_{M\to\infty} L(M,\vec{b}) = p^{1-1/\alpha}$, we have for any $\epsilon \in (0,1)$, there exists M_0, such that, for $M \ge M_0$,

$$P_{-\epsilon}(x;\vec{\mu},\vec{b}) \le P(x;\vec{\mu},\vec{b}) \le P_{\epsilon}(x;\vec{\mu},\vec{b})$$

where

$$P_{\epsilon}(x;\vec{\mu},\vec{b}) \triangleq (1-p)^M + (1+\epsilon)p^{1-1/\alpha}M^{\alpha-1}Q(x),$$
$$P_{-\epsilon}(x;\vec{\mu},\vec{b}) \triangleq (1-p)^M + (1-\epsilon)p^{1-1/\alpha}M^{\alpha-1}Q(x).$$

Without loss of generality, we assume that M can take real values in $P_{\epsilon}(x;\vec{\mu},\vec{b})$ and $P_{-\epsilon}(x;\vec{\mu},\vec{b})$. For a given x, let $M_{\epsilon}^*(x)$ and $M_{-\epsilon}^*(x)$ be the values of M that minimize $P_{\epsilon}(x;\vec{\mu},\vec{b})$ and $P_{-\epsilon}(x;\vec{\mu},\vec{b})$, respectively. Solving $\partial P_{\epsilon}(x;\vec{\mu},\vec{b})/\partial M = 0$ and $\partial P_{-\epsilon}(x;\vec{\mu},\vec{b})/\partial M = 0$, we obtain

$$M_{\epsilon}^*(x) \sim M_{-\epsilon}^*(x) \sim (1-\alpha)\log x/\log(1-p)$$

as $x \to \infty$ for any $\epsilon \in (0,1)$, which completes the proof. □

E. Proof of Theorem 4

Proof. Suppose towards contradictions that there exists $1 \le i < j \le M$ such that $b_i^* = b_j^*$. Then, we consider a new optimization problem as follows.

$$\begin{aligned}\min_{b_i,b_j}\quad & P(x;(0,\cdots,0,1),\vec{b})\\ \text{subject to}\quad & \sum_{m=1}^{M} b_m = 1, \\ & b_m = b_m^*, \qquad 1 \le m \le M, m \ne i,j,\\ & 0 \le b_m \le 1, \qquad 1 \le m \le M.\end{aligned} \tag{26}$$

Recall that b_m^* is the optimal solution of the original problem (4). In this new problem (26), we restrict $b_m = b_m^*$ for all b_m's except b_i and b_j, and minimize the miss probability over b_i and b_j. Thus, b_i^* and b_j^* should also be the optimal solution of Problem (26), and satisfy the optimal condition

$$\left.\frac{\partial P(x;\vec{\mu},\vec{b})}{\partial b_i}\right|_{b_i=b_i^*} = \left.\frac{\partial P(x;\vec{\mu},\vec{b})}{\partial b_j}\right|_{b_j=b_j^*},$$

which is equivalent to

$$\left.\frac{\partial L(M,\vec{b})}{\partial b_i}\right|_{b_i=b_i^*} = \left.\frac{\partial L(M,\vec{b})}{\partial b_j}\right|_{b_j=b_j^*}. \tag{27}$$

Moreover, we have

$$\left.\frac{\partial L(M,\vec{b})}{\partial b_i}\right|_{b_i=(b_i^*+b_j^*)/2} > \left.\frac{\partial L(M,\vec{b})}{\partial b_j}\right|_{b_j=(b_i^*+b_j^*)/2},$$

which indicate that $b_i = b_j = (b_i^*+b_j^*)/2$ does not satisfy (27) and is not the optimal solution of Problem (26). Therefore, we must have $b_i^* \ne b_j^*$ for the original problem. Furthermore, since we assume $b_i \ge b_j$, the proof is completed. □

F. Proof of Lemma 2

Proof. Note that, if the channel reliability levels are all equal, the miss probability of any static policy only depends on the number of replications of each data item and is irrelevant to how these replications are placed on caches. After deciding replication strategy, we can simply store the data item to the first m caches, where m is the number of replications for the item. Therefore, in order to design an optimal static policy that minimizes the miss probability, we only need to find the optimal cache space allocation.

If the data item d_i is stored in m caches, the miss probability of d_i is $(1-p)^m$. Now, if we store one more copy of d_i in the system, the overall miss probability of the system will be reduced by

$$\Delta_i(m) \triangleq q_i((1-p)^m - (1-p)^{m+1}) = q_i p(1-p)^m.$$

In other words, $\Delta_i(m)$ represents the marginal gain to store one more d_i in the system when there are already m copies. The optimal static policy can be found by starting with empty caches and gradually adding items with the largest marginal gains. □

G. Proof of Theorem 5

Proof. Recalling the EA and UA policies and Theorem 1, we have

$$\mathbb{P}_{\text{miss}}^{\text{EA}} \sim (1-p)^M + L\left(M,\vec{b}_{\text{EA}}\right)M^{\alpha-1}Q(x),$$
$$\mathbb{P}_{\text{miss}}^{\text{UA}} \sim (1-p)^M + L\left(M,\vec{b}^{\circ}(x)\right)M^{\alpha-1}Q(x) \quad \text{for } p > p_{\text{th}},$$
$$\mathbb{P}_{\text{miss}}^{\text{UA}} = \mathbb{P}_{\text{miss}}^{\text{EA}} \quad \text{for } p \le p_{\text{th}}.$$

Therefore, we have

$$\frac{\mathbb{P}_{\text{miss}}^{\text{UA}} - (1-p)^M}{\mathbb{P}_{\text{miss}}^{\text{EA}} - (1-p)^M} = \begin{cases} L(M,\vec{b}^{\circ}(x))/L(M,\vec{b}_{\text{EA}}) & \text{for } p > p_{\text{th}},\\ 1 & \text{for } p \le p_{\text{th}}.\end{cases}$$

Moreover, since p_{th} is the unique solution to $L(M,\vec{b}^{\circ}(x)) = L(M,\vec{b}_{\text{EA}})$, the proof is completed. □

H. Proof of Theorem 6

The proof of Theorem 6 is similar to the proof of Theorem 1 (see Section VIII-B). First, we will modify Lemma 4 for heterogenous channels.

Lemma 7. *Consider M caches with heterogenous channels. Assume that the data items are hashed to all caches. Conditional on $\mathcal{J}_0 = \{C_1,\cdots,C_m\}$ (i.e., the first m caches are accessible at τ_0), as the total cache space $x \to \infty$, we have*

$$\mathbb{P}[\textit{Miss for } R_0|\mathcal{J}_0 = \{C_1,\ldots,C_m\},H] \sim Q(\bar{x})$$

where $\bar{x} = \left(\sum_{k=1}^{m} \prod_{j=0}^{k-1}(1-p_j){b_k}^{\alpha}\right)^{1/\alpha} x$.

Proof. The proof is similar to the proof of Lemma 4. Applying (8), we have

$$\begin{aligned}&\mathbb{P}[\text{Miss for } R_0|\mathcal{J}_0 = \{C_1,\ldots,C_m\}, H]\\ &\quad = \mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > T_k^{\leftarrow}(b_k x)\}].\end{aligned}$$

In Step 1 we will estimate $\mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > n_k\}]$, where n_k's are given constants. In Step 2, we will show that $\mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > T_k^{\leftarrow}(b_k x)\}]$ can be approximated by $\mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > \bar{T}_k^{\leftarrow}(b_k x)\}]$, where

$$\bar{T}_k^{\leftarrow}(x) \approx \Gamma(1-1/\alpha)^{-\alpha} c^{-1} {p_k}^{-1} x^{\alpha}.$$

Step 1: For $\forall n \geq 1$, let $Y_i(n) = \sum_{j=1}^{n} \mathbf{1}(\{R_{-j} = d_i\})$, $1 \leq i \leq N$. $Y_i(n)$ represents the number of requests that fetch data d_i during τ_{-n} and τ_{-1}, and follows a binomial distribution. We have,

$$\begin{aligned}&\mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > n\}|R_0 = d_i]\\ &= \sum_{j=0}^{n} \mathbb{P}[Y_i(n) = j]\mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > n\}|Y_i(n) = j]\\ &= \sum_{j=0}^{n} \mathbb{P}[Y_i(n) = j]\Pi_{k=1}^{m}(1-p_k)^j\\ &= \mathbb{E}\left[\exp\left(Y_i(n)\cdot\log\left(\Pi_{k=1}^{m}(1-p_k)\right)\right)\right]\\ &= \mathcal{M}_{Y_i(n)}\left(\log\left(\Pi_{k=1}^{m}(1-p_k)\right)\right)\\ &= \left(1-q_i+q_i\Pi_{k=1}^{m}(1-p_k)\right)^n, \end{aligned} \tag{28}$$

where $\mathcal{M}_{Y_i(n)}(t) \triangleq \mathbb{E}[\exp(tY_i(n))]$ is the moment generating function of $Y_i(n)$.

By combining (9) and (28), we have

$$\begin{aligned}\mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > n_k\}] &= \sum_{i=1}^{\infty} q_i \mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > n_k\}|R_0 = d_i]\\ &= \sum_{i=1}^{\infty} q_i \Pi_{k=1}^{m}\left(1-q_i+q_i\Pi_{j=1}^{k}(1-p_j)\right)^{n_k - n_{k+1}}.\end{aligned}$$

Using the same technique that proves (12), we can show that as $n_m \to \infty$,

$$\begin{aligned}&\mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > n_k\}]\\ &\sim \sum_{i=1}^{\infty} q_i \exp\left(-q_i\sum_{k=1}^{m}(1-\Pi_{j=1}^{k}(1-p_j))(n_k - n_{k+1})\right)\\ &\sim \frac{c^{1/\alpha}\Gamma(2-1/\alpha)}{\alpha-1}\left(p\sum_{k=1}^{m}\Pi_{j=0}^{k-1}(1-p_j)n_k\right)^{-1+1/\alpha}.\end{aligned} \tag{29}$$

Step 2: Define

$$\begin{aligned}\bar{T}_m(n) &\triangleq \sum_{i=1}^{\infty}\left(1-(1-q_i+q_i(1-p_m))^n\right)\\ \bar{T}_m^{\leftarrow}(x) &\triangleq \min\{n : \bar{T}_m(n) \geq x\}.\end{aligned}$$

We have, as $n \to \infty$

$$\bar{T}_m(n) \sim \sum_{i=1}^{\infty}\left(1-e^{-q_i p_m n}\right) \sim \Gamma\left(1-\frac{1}{\alpha}\right)(cnp_m)^{1/\alpha},$$

and as $x \to \infty$

$$\bar{T}_m^{\leftarrow}(x) \sim \Gamma(1-1/\alpha)^{-\alpha} c^{-1} {p_m}^{-1} x^{\alpha}.$$

Then, applying (29) and using the same technique that proves (20) and (19), we can show that, as $x \to \infty$,

$$\begin{aligned}&\mathbb{P}[\text{Miss for } R_0|\mathcal{J}_0 = \{C_1,\ldots,C_m\}, H]\\ &= \mathbb{P}[\cap_{k=1}^{m}\{\sigma_k > T_k^{\leftarrow}(b_k x)\}]\\ &\sim \frac{c^{1/\alpha}\Gamma(2-1/\alpha)}{\alpha-1}\left(p\sum_{k=1}^{m}\Pi_{j=0}^{k-1}(1-p_j)T_k^{\leftarrow}(b_k x)\right)^{-1+1/\alpha}\\ &\sim \frac{c^{1/\alpha}\Gamma(2-1/\alpha)}{\alpha-1}\left(p\sum_{k=1}^{m}\Pi_{j=0}^{k-1}(1-p_j)\bar{T}_k^{\leftarrow}(b_k x)\right)^{-1+1/\alpha}\\ &\sim \frac{\Gamma(1-1/\alpha)^{\alpha}}{\alpha}\frac{c}{\bar{x}^{\alpha-1}} = Q(\bar{x}),\end{aligned}$$

where $\bar{x} = \left(\sum_{k=1}^{m} \prod_{j=0}^{k-1}(1-p_j){b_k}^{\alpha}\right)^{1/\alpha} x$. □

Proof of Theorem 6. Using a similar technique that proves Theorem 1, we have,

$$\begin{aligned}&\mathbb{P}[\text{Miss for } R_0|H]\\ &= \sum_{m=1}^{M}\sum_{k=1}^{\binom{M}{m}}\mathbb{P}[\mathcal{I}_0 = \mathcal{S}_k^{(m)}]\mathbb{P}[\text{Miss for } R_0|\mathcal{I}_0 = \mathcal{S}_k^{(m)}, H]\\ &= \sum_{m=1}^{M}\sum_{k=1}^{\binom{M}{m}}\mathbb{P}[\mathcal{I}_0 = \mathcal{S}_k^{(m)}]\Big(\Pi_{i=1}^{m}(1-p_{I_{k,i}^{(m)}})\\ &\quad + \sum_{(i,j):\mathcal{S}_i^{(j)}\subseteq\mathcal{S}_k^{(m)}}\Big(\mathbb{P}[\mathcal{J}_0 = \mathcal{S}_i^{(j)}|\mathcal{I}_0 = \mathcal{S}_k^{(m)}, H]\\ &\qquad \cdot \mathbb{P}[\text{Miss for } R_0|\mathcal{J}_0 = \mathcal{S}_i^{(j)}, \mathcal{I}_0 = \mathcal{S}_k^{(m)}, H]\Big)\Big).\end{aligned}$$

Then, applying Lemmas 5, 6 and 7 finishes the proof of Theorem 6. □

I. Proof of Lemma 3

Different from the static policy for homogenous channels, the miss probability depends not only on the number of replications of each data, but also on which caches the data are stored, when the channels have heterogenous reliability levels. Assume without loss of generality that p_m's are non-increasing with respect to m. If we decide to store m replications of a data item in the system, they should be stored in the most reliable caches (i.e., $C_1, C_2, \cdots, C_m$) to minimize the miss probability. In other words, given the replication strategy, the data placement is fixed. Therefore, we only need to decide the cache size of each involved caches.

Similar to the homogenous case, for $m \geq 0$, we define the marginal gain $\Delta_i(m)$ as the reduction in the overall miss probability by storing one more d_i in the system when there are already m copies, i.e.,

$$\begin{aligned}\Delta_i(m) &= q_i\left(\prod_{k=0}^{m}(1-p_k) - \prod_{k=0}^{m+1}(1-p_k)\right)\\ &= q_i p_{m+1}\prod_{k=0}^{m}(1-p_k).\end{aligned}$$

The optimal static policy can be obtained by starting with empty caches and gradually adding items with the largest marginal gains.

REFERENCES

[1] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.*, vol. 1. IEEE, 1999, pp. 126–134.

[2] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's highly available key-value store," in *Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles*, ser. SOSP '07. ACM, 2007, pp. 205–220.

[3] "Memcached," http://memcached.org/.

[4] "Redis," http://redis.io/.

[5] "Aerospike," http://www.aerospike.com/.

[6] N. Megiddo and D. S. Modha, "ARC: A self-tuning, low overhead replacement cache." in *FAST*, vol. 3, no. 2003, 2003, pp. 115–130.

[7] S. Jiang and X. Zhang, "LIRS: an efficient low inter-reference recency set replacement policy to improve buffer cache performance," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1, pp. 31–42, 2002.

[8] A. S. Tanenbaum, *Modern Operating Systems*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2001.

[9] "MySQL," https://www.mysql.com/.

[10] J. Tan, G. Quan, K. Ji, and N. Shroff, "On resource pooling and separation for LRU caching," in *Proceedings of the 2018 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*. ACM, 2018.

[11] W. Chu, M. Dehghan, D. Towsley, and Z.-L. Zhang, "On allocating cache resources to content providers," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. ACM, 2016, pp. 154–159.

[12] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, and M. Paleczny, "Workload analysis of a large-scale key-value store," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1, pp. 53–64, Jun. 2012.

[13] W. Chu, M. Dehghan, J. C. Lui, D. Towsley, and Z.-L. Zhang, "Joint cache resource allocation and request routing for in-network caching services," *Computer Networks*, vol. 131, pp. 1–14, 2018.

[14] R. Nishtala, H. Fugal, S. Grimm, M. Kwiatkowski, H. Lee, H. C. Li, R. McElroy, M. Paleczny, D. Peek, P. Saab *et al.*, "Scaling Memcache at Facebook." in *nsdi*, vol. 13, 2013, pp. 385–398.

[15] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, "Fog computing: Focusing on mobile users at the edge," *arXiv preprint arXiv:1502.01815*, 2015.

[16] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[17] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[18] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.

[19] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web," in *Proceedings of the 1997 ACM Symposium on Theory of Computing*. ACM, 1997, pp. 654–663.

[20] R. Fagin, "Asymptotic miss ratios over independent references," *Journal of Computer and System Sciences*, vol. 14, no. 2, pp. 222–250, 1977.

[21] P. R. Jelenković, "Asymptotic approximation of the move-to-front search cost distribution and least-recently-used caching fault probabilities," *The Annals of Applied Probability*, no. 2, pp. 430–464, 1999.

[22] P. R. Jelenković and X. Kang, "LRU caching with moderately heavy request distributions," in *2007 Proceedings of the Fourth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*. SIAM, 2007, pp. 212–222.

[23] G. Quan, K. Ji, and J. Tan, "LRU caching with dependent competing requests," in *2018 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2018.

[24] R. Hirade and T. Osogami, "Analysis of page replacement policies in the fluid limit," *Operations Research*, vol. 58, no. 4-part-1, pp. 971–984, 2010.

[25] N. Gast and B. Van Houdt, "Transient and steady-state regime of a family of list-based cache replacement algorithms," *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, no. 1, pp. 123–136, 2015.

[26] K. Ji, G. Quan, and J. Tan, "Asymptotic miss ratio of LRU caching with consistent hashing," in *2018 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2018.

[27] J. Li, T. K. Phan, W. Chai, D. Tuncer, G. Pavlou, D. Griffin, and M. Rio, "Dr-cache: Distributed resilient caching with latency guarantees," in *2018 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2018.

[28] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal routing and content caching in heterogeneous networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015.

[29] S. Ioannidis and E. Yeh, "Adaptive caching networks with optimality guarantees," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 44, no. 1. ACM, 2016, pp. 113–124.

[30] G. Quan, J. Tan, and A. Eryilmaz, "Counterintuitive characteristics of optimal distributed LRU caching over unreliable channels," in *2019 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2019, pp. 694–702.

[31] D. S. Berger, R. K. Sitaraman, and M. Harchol-Balter, "Adaptsize: Orchestrating the hot object memory cache in a content delivery network." in *NSDI*, 2017, pp. 483–498.

[32] D. S. Berger, N. Beckmann, and M. Harchol-Balter, "Practical bounds on optimal caching with variable object sizes," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 2, p. 32, 2018.

[33] F. Chung, F. R. Chung, F. C. Graham, L. Lu, K. F. Chung *et al.*, *Complex graphs and networks*. American Mathematical Soc., 2006, no. 107.

Guocong Quan received the B.E. degree in electrical engineering from the University of Science and Technology of China in 2016. He is currently pursuing the Ph.D. degree in electrical and computer engineering at The Ohio State University. His research interest focuses on resolving challenges in distributed caching systems. He received the 2019 IEEE Infocom Best Paper Award.

Jian Tan received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2002, and the M.S. and Ph.D. degrees from Columbia University, New York, NY, USA, in 2004 and 2008, respectively, all in electrical engineering. From 2009 to 2010, he was a Postdoctoral Researcher with the Networking and Communications Research Lab, The Ohio State University, Columbus, OH, USA. From 2010 to 2015, he worked at IBM T. J. Watson Research Center, Yorktown Heights, NY, and the Institute of Data Science and Technology of Alibaba, Seattle, WA. Since 2016, he has been an assistant professor and then holds an adjunct position of electrical and computer engineering with The Ohio State University. His research interests focus on stochastic modeling and statistical algorithms for networks and distributed systems. He won the Eliahu Jury Award from Columbia University for his Ph.D. thesis work, and is a coauthor of the 2007 ITC Best Student Paper, 2007 DCOSS Best Paper, 2013 IFIP Performance Best Student Paper and 2019 IEEE Infocom Best Paper.

Atilla Eryilmaz (S'00–M'06–SM'17) received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 2001 and 2005, respectively. From 2005 to 2007, he was a Post-Doctoral Associate with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology. Since 2007, he has been with The Ohio State University, where he is currently a Professor of electrical and computer engineering. His research interests include design and analysis for communication networks, optimal control of stochastic networks, optimization theory, distributed algorithms, pricing in networked systems, and information theory. He received the NSF-CAREER Award in 2010 and two Lumley Research Awards for Research Excellence in 2010 and 2015. He is a coauthor of the 2012 IEEE WiOpt Conference Best Student Paper, and subsequently received the 2016 IEEE Infocom, 2017 IEEE WiOpt, 2018 IEEE WiOpt, and 2019 IEEE Infocom Best Paper Awards. He served as a TPC Co-Chair of the IEEE WiOpt in 2014 and the ACM Mobihoc in 2017. He served as an Associate Editor (AE) for the IEEE/ACM Transactions on Networking from 2015 to 2019. He has been an AE of the IEEE Transactions on Network Science and Engineering since 2017.