Minimizing Edge Caching Service Costs Through Regret-Optimal Online Learning

Guocong Quan, Atilla Eryilmaz, *Senior Member, IEEE*, Ness B. Shroff, *Fellow, IEEE*

***Abstract*—Edge caching has been widely implemented to efficiently serve data requests from end users. Numerous edge caching policies have been proposed to adaptively update the cache contents based on various statistics. One critical statistic is the miss cost, which could measure the latency or the bandwidth/energy consumption to resolve the cache miss. Existing caching policies typically assume that the miss cost for each data item is fixed and known. However, in real systems, they could be random with unknown statistics. A promising approach would be to use online learning to estimate the unknown statistics of these random costs, and make caching decisions adaptively. Unfortunately, conventional learning techniques cannot be directly applied, because the caching problem has additional cache capacity and cache update constraints that are not covered in traditional learning settings. In this work, we resolve these issues by developing a novel edge caching policy that learns uncertain miss costs efficiently, and is shown to be asymptotically optimal. We first derive an asymptotic lower bound on the achievable regret. We then design a Kullback-Leibler lower confidence bound (KL-LCB) based edge caching policy, which adaptively learns the random miss costs by following the "optimism in the face of uncertainty" principle. By employing a novel analysis that accounts for the new constraints and the dynamics of the setting, we prove that the regret of the proposed policy matches the regret lower bound, thus showing asymptotic optimality. Further, via numerical experiments we demonstrate the performance improvements of our policy over natural benchmarks.**

I. Introduction

Edge caching has been widely implemented to store data items closer to end users and accelerate data access. It is reported that about 50% photo traffic on Facebook are served by geographically distributed edge caches [1]. Edge caches typically have limited capacity and can only accommodate a small fraction of the entire dataset. When the requested data item is not stored in the edge cache, we call it a cache miss and the data item has to be fetched from the backend data storage to serve the request, which will incur a large delay and consume more bandwidth or power resources. A critical question for caching design is *which data items should be stored in the edge cache?*

Numerous caching policies have been designed to update the cache content based on different data statistics. One critical statistic is the miss cost, i.e., the cost to fetch the requested data from other storage when it is not stored in the edge cache. The miss cost is a general concept depending on the specific application (e.g., the cost could represent the latency or bandwidth/energy consumption required to fetch the missed data from backend). Intuitively, we should cache data items that may potentially incur larger miss costs, so that the expected cost to serve the request is minimized. By following this principle, various caching policies have been proposed [2], [3], [4], [5], [6]. However, almost all of them assume that the miss costs are fixed and known, which is not the case in real systems. The miss cost of a data item could be *random with unknown statistics* in real systems. For example, the miss cost may depend on the geographic locations of the backend storage that sends the missed data back, communication environments, network traffic flows, etc. Existing caching policies cannot satisfactorily handle such uncertainty. To fill this gap, we develop new edge caching policies that learn the unknown statistics of the random miss costs adaptively and efficiently.

A promising approach is to use online learning to estimate these unknown statistics of miss costs. However, existing online learning approaches cannot be directly applied due to the following reasons. 1) The learning actions and the caching decisions are correlated and should be jointly optimized. Specifically, we can observe samples for the uncertain miss costs, only when the corresponding data item is not stored in the edge cache. 2) Caching problems have additional cache capacity and content update constraints that are not covered in the traditional online learning settings. For example, the cache contents in the next time slot will remain the same as the current one, if there are no cache updates, which naturally introduces time correlations. Due to the dependency between learning and caching decisions, such time correlations will exist in the action sets of the learning process. These constraints make the problem highly non-trivial. We show in Section III that a heuristic design could almost always make wrong caching decisions and achieves poor performance.

We address these challenges by designing a novel edge caching policy that learns the unknown statistics of miss costs efficiently. In particular, we first characterize the best achievable caching performance by establishing a regret lower bound. Inspired by the "optimism in the face of uncertainty"

This work has been supported in part by NSF grants: CNS-2312836, CNS-2223452, CNS-2225561, CNS-2112471, CNS-2106933, CNS-2106679, CNS-2007231, CNS-1824337, a grant from the Office of Naval Research: N00014-19-1-2621, grants from the Army Research Office: W911NF-21-1-0244, W911NF-24-1-0103, and was sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF23-2-0225. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Guocong Quan and Atilla Eryilmaz are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: quan.72@osu.edu; eryilmaz.2@osu.edu). Ness B. Shroff is with the Department of Electrical and Computer Engineering, and the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: shroff.11@osu.edu).

principle in learning literature [7], [8], we then propose a novel KL-LCB based edge caching policy that adaptively learns the unknown statistics of the miss costs. In order to analyze the theoretical performance of the proposed policy, we are required to prove almost-sure convergence results for critical caching statistics, which are not covered by traditional online learning analysis. Based on these new results, we prove that the proposed policy achieves the regret lower bound, and is therefore asymptotically optimal. Our key contributions are summarized as follows.

- We reveal the non-triviality of learning miss costs in caching systems. We introduce a heuristic learning design and carefully explain that it could achieve significant inefficiency (see Section III).
- We derive a regret lower bound for any "good" polices (see Section IV), and develop an asymptotically optimal KL-LCB based edge caching policy that achieves this regret lower bound (see Section V). The analysis for the proposed policy employs novel ideas to deal with the new constraints and dynamics in caching systems, and could be potentially leveraged to analyze learning mechanisms for other systems.
- We conduct extensive numerical experiments to evaluate the proposed KL-LCB based edge caching policy, and compare it with a few benchmarks. It is shown that the proposed policy achieves significantly better performance than the other benchmarks (see Section VII).

Related Works: Cost-based caching policies have been extensively studied. The GreedyDual policy evicts the data item with the smallest miss cost when the cache is full [5]. Different designs have been proposed to implement the Greedy-Dual policy [4], [3]. The GreedyDual-Size policy considers data items with different sizes and uses costs per unit data size as a critical factor for caching update [9]. It is further generalized as the Greedy-Dual-Size-Frequency (GDSF) policy to make caching decisions based on the joint effect of data frequency, sizes and costs [2]. Specifically, the GDSF policy attempts to cache the data items with large *frequency* × *cost / size* values. In [6], Hyperbolic caching is proposed to provide flexible caching service for web applications, and is implemented in real systems such as Redis and Django. It prioritizes data items based on a general function that could depend on miss costs, expiration times, windowing, etc. Other factors (e.g., freshness, latency) are also considered in cost-based policies designed for a variety of applications [10], [11], [12]. Notably, these works assume that the miss costs are known. For unknown miss costs, efficient cost-learning mechanisms jointly optimized with caching decisions are needed.

Leveraging online learning techniques to improve caching performance has received more and more attention. However, most of the existing works in this area focus on learning unknown data popularities or user preference [13], [14], [15], [16], [17], [18]. Learning data popularities has different constraints and dynamics from learning miss costs. And therefore, these approaches cannot be extended to directly solve the cost-learning problem. In [19], fetching costs are considered for caching at small base stations. The paper first assumes known cost distributions and develop efficient algorithms to solve the cost minimization problem. Then, unknown cost distributions are considered and a Q-learning based approach is proposed to estimate the unknown cost distributions. However, no theoretical performance guarantee is provided for this approach.

We also note that edge caches are typically connected with end users through wireless channels. The unreliability and broadcasting capability of wireless channels have introduced challenges as well as opportunities for edge caching optimization [20], [21], [22], [23], [24], [25], [26]. However, in this paper, we do not consider such dynamics of wireless channels, which deserves a deep dive in future research.

II. Problem Formulation

In this section, we first introduce the system model for edge caching with uncertain miss costs. Next, we formulate a service cost minimization problem for solving the optimal edge caching policy. We then define the regret to measure the performance of an edge caching policy.

A. System Model

Consider an edge caching scenario as illustrated in Fig. 1. Edge caches are placed at the network edges (e.g., on base stations) and are the closest to the end users. Edge caches have very limited cache capacity and can only store a small fraction of the entire dataset. The backend data centers are in the core of the network and stores the entire dataset. On the route from the edge to the backend data center, some intermediate nodes may have caching capabilities, which store part of the dataset and are closer to the network edge than the data center. The considered storage architecture is commonly used in today's content delivery networks [1]. In this paper, we focus on a single edge cache and investigate how to develop an optimal content update policy for it.

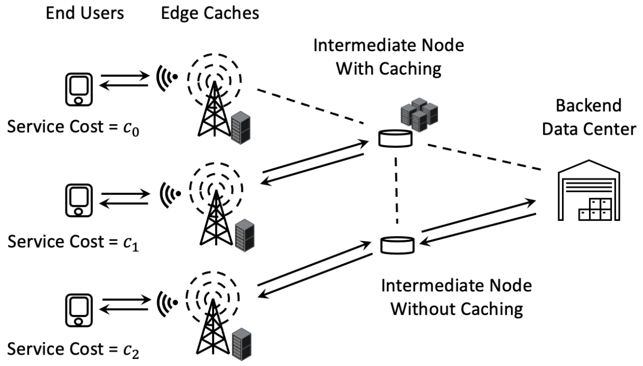


Fig. 1: Edge caching with disparate service costs.

We consider a discrete-time system with time $t = 1, 2, \cdots$. Let $\mathcal{D} = \{d_1, d_2, \cdots, d_N\}$ be a set of N data items, where the item sizes are assumed to be 1[1]. Let K be the capacity of the edge cache, $1 \leq K < N$. At each time slot t, a data request is generated and sent to the edge cache. Let R_t denote the data item that is requested at time t, $R_t \in \mathcal{D}$. We characterize R_t by

[1] In Section VI-A, we will discuss how to generalize our main results to allow for non-identical data sizes.

the independent reference model (IRM), which is commonly assumed in caching analysis [27], [28]. Specifically, the data requests are independently generated based on the unknown popularity $p_i \triangleq \mathbb{P}[R_t = d_i]$, with $p_i \in (0,1)$ and $\sum_{i=1}^{N} p_i = 1$.

Once a data request is received, we will first try to serve the request from the edge cache. If the requested data is stored in the edge cache (i.e., a cache hit), we can serve the request immediately and pay a negligible hit cost c_0, as shown in Fig. 1. Instead, if the data is not stored in the edge cache (i.e., a cache miss), we need to fetch the data from the data storage in the network core. Specifically, we can fetch the missed data from the intermediate caching node at a relatively small cost c_1, if it is stored there. Otherwise, the data has to be retrieved from the backend data center at a larger cost c_2. After retrieving the missed data, the request can be served without loading the missed data into the edge cache.

However, whether the missed data can be served from an intermediate caching node is uncertain because 1) the data stored in the intermediate node are dynamically changing and 2) the dynamic routing strategies may not always direct the request to the same intermediate node depending on real time network traffic. To model such uncertainties, we use Bernoulli random variables with unknown parameters to represent the miss costs. In particular, when a cache miss happens, the service cost of the data item d_i, $1 \le i \le N$, takes value c_2 with a probability q_i and c_1 with a probability $1 - q_i$ independently in each time slot, where q_i can be interpreted as the probability that the data item d_i is not stored in the intermediate caching node. We assume that the constant c_2, c_1 and c_0 are known and satisfy $c_2 > c_1 > c_0 \ge 0$. The parameter q_i's, as well as the data popularity p_i's, are unknown and need to be estimated for designing efficient edge caching policies.

B. Edge Caching for Cost Minimization

In this paper, we focus on the edge caching problem without prefetching, i.e., an uncached data item will not be loaded into the cache unless a user request it. This is because prefetching operations will incur additional but unnecessary data fetching cost in our model. In particular, an edge caching policy needs to make the following decision:

- When a cache miss happens, should the requested data be loaded into the edge cache?
- If the cache is already full, which cached item should be evicted to make room for the new one?

Our goal is to design an efficient edge caching policy to minimize the accumulated expected service costs.

For a time horizon n, let $Cost(n)$ denote the expected service cost accumulated from $t = 1$ to $t = n$. For each data item d_i, define

$$T_i^{\text{in}}(n) = \sum_{t=1}^{n} \mathbf{1}(d_i \text{ is stored in the edge cache at time } t),$$

$$T_i^{\text{out}}(n) = \sum_{t=1}^{n} \mathbf{1}(d_i \text{ is not stored in the edge cache at time } t).$$

$T_i^{\text{in}}(n)$ and $T_i^{\text{out}}(n)$ are random variables and depend on the edge caching policy. Let

$$\gamma_i = q_i c_2 + (1 - q_i) c_1 - c_0,$$

which can be interpreted as the cost reduction achieved by storing d_i in the edge cache. We can derive the accumulated expected cost as

$$\begin{aligned} Cost(n) =& \sum_{i=1}^{N} \mathbb{E}[T_i^{\text{out}}(n)] p_i (\gamma_i + c_0) + \sum_{i=1}^{N} \mathbb{E}[T_i^{\text{in}}(n)] p_i c_0 \\ =& n c_0 + \sum_{i=1}^{N} \mathbb{E}[T_i^{\text{out}}(n)] p_i \gamma_i, \end{aligned}$$

where the last equation holds because $T_i^{\text{in}}(n) + T_i^{\text{out}}(n) = n$ for each data item d_i. Our objective is to design good edge caching polices that minimize $Cost(n)$ for large n.

First, we consider an idealized scenario where the parameter p_i and q_i are known. In this scenario, the optimal policy is to always store the data items with the largest $p_i \gamma_i$ values in the edge cache. And we denote this optimal policy as π^*. This strategy attempts to achieve the largest cost reduction through caching. Without loss of generality, we assume that the data items could be strictly ordered based on $p_i \gamma_i$ and are indexed such that $p_i \gamma_i > p_j \gamma_j$ for any $1 \le i < j \le N$. The results of this paper can be easily extended to the case without the strict ordering (i.e., existing $i \neq j$ with $p_i \gamma_i = p_j \gamma_j$). Then, the optimal policy π^* simply stores the first K items. We call the set $\{d_i : 1 \le i \le K\}$ the optimal choice set, and the set $\{d_i : K+1 \le i \le N\}$ the suboptimal choice set.

In this paper, we consider a more realistic setting where q_i and p_i are unknown for reasons that are described in the previous section. Consequently, γ_i is also unknown, being a function of q_i. Thus, we need to adaptively learn p_i and q_i and update the cache content accordingly.

Due to the embedded learning nature of the problem, the performance of edge caching policies will be evaluated by regrets. This is a classical learning metric which characterizes the difference between the expected accumulated cost achieved by an edge caching policy and the one achieved by the idealized optimal policy π^* with known p_i and q_i. Formally, we define the regret over a time horizon n as

$$\begin{aligned} & Regret(n) \\ &= Cost(n) - \left(\sum_{i=1}^{K} n p_i c_0 + \sum_{i=K+1}^{N} n p_i (q_i c_2 + (1 - q_i) c_1) \right) \\ &= Cost(n) - \left(n c_0 + \sum_{i=K+1}^{N} n p_i \gamma_i \right) \\ &= \sum_{i=1}^{N} \mathbb{E}[T_i^{\text{out}}(n)] p_i \gamma_i - \sum_{i=K+1}^{N} n p_i \gamma_i. \end{aligned}$$

Minimizing the expected accumulated cost $Cost(n)$ is equivalent to minimizing the regret $Regret(n)$. Next, we will introduce our motivation to use online learning techniques and the challenges.

III. Motivations and Challenges

In this section, we first introduce our motivation by showing that a natural heuristic design could achieve significant inefficiency. To solve this issue, we propose to leverage online learning to adaptively estimate the unknown parameters. However, existing online learning algorithms do not consider the specific constraints and dynamics in caching systems and therefore cannot be applied directly. We then introduce the challenges of combining online learning and caching.

A. Motivation: Inefficiency of Heuristic Designs

An edge caching policy needs to estimate the unknown parameters p_i and q_i based on history information and make caching decisions accordingly. Define

$$T_i^{\text{miss}}(t) = \sum_{s=1}^{t} \mathbf{1}(R_s = d_i \text{ and is not in edge cache}),$$

$$T_i^{\text{back}}(t) = \sum_{s=1}^{t} \mathbf{1}(R_s = d_i \text{ and is served from}$$
$$\text{the backend data center}).$$

Recall that q_i is the probability that the data item d_i is not stored in the intermediate cache. The unbiased estimation of p_i and q_i at time t could be the sample means

$$\hat{p}_i(t) = \sum_{s=1}^{t} \mathbf{1}(d_i \text{ is requested at } s)/t, \tag{1}$$

$$\hat{q}_i(t) = T_i^{\text{back}}(t)/T_i^{\text{miss}}(t). \tag{2}$$

A heuristic caching policy would estimate γ_i by

$$\hat{\gamma}_i(t) = \hat{q}_i(t)c_2 + (1 - \hat{q}_i(t))\, c_1 - c_0, \tag{3}$$

and evict the data item with the smallest $\hat{p}_i(t)\hat{\gamma}_i(t)$ value when the cache is full. We formally describe this heuristic edge caching policy in Algorithm 1.

Algorithm 1: Heuristic Edge Caching Policy

```
Initialization: $T_i^{\text{miss}}(0) = T_i^{\text{back}}(0) = \hat{q}_i(0) = 0$,
  $\hat{\gamma}_i(0) = c_1 - c_0$, $1 \le i \le N$;
for $t = 1 : n$ do
    Assume w.o.l.g. that $R_t = d_i$;
    if $d_i$ is not stored in the edge cache then
        Fetch $d_i$ to serve the request;
        Update $\hat{p}_i(t)$ and $\hat{\gamma}_i(t)$ based on (1) and (3);
        if Edge cache is full then
            $j = argmin\{\hat{p}_k(t)\hat{\gamma}_k(t)$ :
              $d_k$ is currently stored in the edge cache$\}$;
            if $\hat{p}_i(t)\hat{\gamma}_i(t) > \hat{p}_j(t)\hat{\gamma}_j(t)$ then
                Load $d_i$ into the cache and evict $d_j$;
        else
            Load $d_i$ into the cache;
        end
end
```

Interestingly, this heuristic policy could be extremely inefficient, which is validated by simulations in Section VII. The issue comes from the estimation of q_i. When a data item d_i is stored in the edge cache, we will not be able to observe its miss cost, and therefore cannot update the estimation of q_i. The inaccurate estimation at the early stage could make the edge cache stop collecting observations for already cached data items and get stuck in a suboptimal solution. In order to solve this issue, we are motivated to leverage online learning techniques to estimate q_i and update cache content strategically.

B. Our Approach: Adaptive Caching via Online Learning

We first highlight an exploration and exploitation trade-off for the proposed edge caching problem:

Exploration: we would like to introduce cache misses intentionally to collect more observations on the miss cost. This could improve the accuracy of the estimations of q_i.

Exploitation: we would like to exploit the current estimation and cache the items with the largest potential gains. This could reduce the overall service costs.

The proposed heuristic policy only exploits but never explores, and therefore performs quite poorly. In order to efficiently learn the unknown parameters and meanwhile achieve good caching performance, we have to balance the exploration and exploitation trade-off.

This trade-off has been extensively studied in online learning literature (in particular, the stochastic multi-armed bandit (MAB) problems [7]). We point out that the proposed edge caching problem share some similarities with combinatorial multi-armed bandit (CMAB) problems. Specifically, it is similar to stochastic combinatorial semi bandits with N arms in total and $N - K$ arms played at each time slot [7]. However, conventional algorithms and analysis for CMAB problems cannot be directly applied to the edge caching scenario due to additional cache capacity and cache update constraints.

C. Key Challenge: Learning with Caching Constraints

The edge caching problem has the following additional constraints compared with the combinatorial bandit problem:

- At each time slot, the edge caching policy can add at most one data item into the cache, which depends on the data requests, while the action space of conventional bandit problems typically has no such constraints.
- The data items that can be stored at time t depend on the cache content at the previous time slot $t - 1$. In contrast, the action space of bandit problems typically does not have such time correlations.

These constraints in edge caching systems make the problem more challenging. The standard bandit algorithms and analysis cannot be directly applied. To solve this issue, we need to propose novel edge caching policies and use new theoretical tools to analyze its performance. To that end, in the next section, we derive a regret lower bound over all policies of interest, followed in the subsequent section by a new design with a novel regret upper bound that matches the scaling of the lower bound, thereby proving its asymptotic-optimality.

IV. Regret Lower Bound

Before designing edge caching polices, we first derive a lower bound for the regret performance of all "good" policies. Following the approach of the seminal work [29], we say that an edge caching policy is a uniformly good policy, if the regret achieved by it satisfies $Regret(n) = o(n^\alpha)$ for $\forall \alpha > 0$. Let $D_{KL}(p,q)$ denote the Kullback-Leibler divergence for two Bernoulli random variables with parameter p and q, respectively. We have for $0 < p < 1$ and $0 < q < 1$,

$$D_{KL}(p,q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}.$$

We prove a regret lower bound in the following theorem.

Theorem 1. *For any uniformly good policy, the regret satisfies*

$$\liminf_{n\to+\infty}\frac{Regret(n)}{\log n} \geq \sum_{i\in\mathcal{S}}\frac{p_i\gamma_i - p_{K+1}\gamma_{K+1}}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1}-p_i(c_1-c_0)}{p_i(c_2-c_1)}\right)\cdot p_i},$$

where $\mathcal{S} = \{1 \le i \le K : p_i(c_1-c_0) < p_{K+1}\gamma_{K+1}\}$.

The proof of this theorem is not a simple application of the proof for classic MAB problems, due to the data request dynamics and cache capacity and update constraints of the edge caching systems. Novel approaches are proposed to capture the structure of the edge caching system, which are presented in Section A. Theorem 1 shows that the regret increases at least logarithmically with the particular coefficient on the right-hand side asymptotically with the time horizon n, if the set $\mathcal{S}$ is not empty. Recall that the data items are indexed such that $p_i\gamma_i$ is decreasing. We make a few remarks on this result:

- The constant on the right-hand side of the regret lower bound depends on the "distance", as measured by the $p_i\gamma_i$ values and a specific form of the Kullback-Leibler divergence, between the data items in the optimal choice set (i.e, $d_i, 1 \le i \le K$) and the best data item in the suboptimal choice set (i.e., d_{K+1}).
- The scaling is independent of N or $N-K$. This is different from most MAB regret performances where the number of arms is a scaling factor. This property arises due to the particular structure of the caching system and is explained in Section VI-C.

Moreover, note that for $d_i \notin \mathcal{S}$ with $1 \le i \le K$ and d_j with $K+1 \le j \le N$, we must have

$$p_i\gamma_i > p_i(c_1-c_0) \ge p_{K+1}\gamma_{K+1} \ge p_j\gamma_j,$$

for $\forall q_i \in (0,1)$, which indicates that we could easily distinguish d_i from the suboptimal choice set (i.e., d_j, $K+1 \le j \le N$), even when the estimation of q_i is arbitrarily chosen. Thus, the scenario with empty $\mathcal{S}$ degenerates to a trivial problem and is not the main focus of this paper.

V. Asymptotically Optimal Edge Caching Policy

In this section, we first propose a novel edge caching policy that leverages online learning ideas. Then, we prove that the proposed policy achieves asymptotically optimal regrets.

A. KL-LCB Based Edge Caching Policy

Instead of estimating q_i by the sample mean $\hat{q}_i(t)$ like the heuristic policy, we follow the principle of "optimism in the face of uncertainty" [7] and estimate q_i by

$$\tilde{q}_i(t) = \min\left\{q \in (0,1) : D_{KL}(\hat{q}_i(t), q) \le \frac{\log f(t)}{T_i^{\text{miss}}(t)}\right\}, \quad (4)$$

where $\hat{q}_i(t)$ is defined in (2) and $f(t) = 1 + t(\log t)^2$. We have $0 < \tilde{q}_i(t) \le \hat{q}_i(t)$ and the "distance" between $\tilde{q}_i(t)$ and $\hat{q}_i(t)$ is characterized by $\log f(t)/T_i^{\text{miss}}(t)$. This design is inspired by the KL-UCB algorithm for reward maximization in conventional stochastic MAB problems [30], [31]. For the edge caching problem, our goal is cost minimization. So we apply the KL-LCB based design that is symmetric to KL-UCB.

With $\tilde{q}_i(t)$, we can estimate γ_i by

$$\tilde{\gamma}_i(t) = \tilde{q}_i(t)c_2 + (1-\tilde{q}_i(t))\,c_1 - c_0, \quad (5)$$

and attempt to cache the items with the largest $\hat{p}_i(t)\tilde{\gamma}_i(t)$. Formally, a KL-LCB based edge caching policy is proposed in Algorithm 2.

When the data request is a cache hit, we do not update the cache content. When the requested data (e.g., d_i) is not stored in the edge cache, we will fetch d_i from the intermediate cache, or, if it is not stored there, from the backend data center, and update the estimations based on the observation. Then, we find the cached data item d_j with the smallest $\hat{p}_j(t)\tilde{\gamma}_j(t)$ values among all cached data items. If $\hat{p}_i(t)\tilde{\gamma}_i(t) > \hat{p}_j(t)\tilde{\gamma}_j(t)$, d_j will be replaced by the newly-requested data item d_i. On the one hand, the proposed policy attempts to cache the data items with large $\hat{p}_i(t)\tilde{\gamma}_i(t)$ values, which exploits the current knowledge. On the other hand, assume d_i is currently cached. $\log(t)/T_i^{\text{miss}}(t)$ will increase with time t, and consequently, $\hat{p}_i(t)\tilde{\gamma}_i(t)$ will gradually decrease. As a result, d_i will finally be evicted, which encourages exploration.

Note that, in Algorithm 2, p_i is simply estimated by the sample mean rather than KL-LCB based statistics. This is because we could always observe the requested data regardless of the caching decisions.

Algorithm 2: KL-LCB based Edge Caching Policy

```
Initialization: $T_i^{\text{miss}}(0) = T_i^{\text{back}}(0) = \hat{q}_i(0) = \tilde{q}_i(0) = 0$,
  $\tilde{\gamma}_i(0) = c_1 - c_0$, $1 \le i \le N$;
for $t = 1 : n$ do
    Assume w.o.l.g. that $R_t = d_i$;
    if $d_i$ is not stored in the edge cache then
        Fetch $d_i$ to serve the request;
        Update $\hat{p}_i(t)$ and $\tilde{\gamma}_i(t)$ based on (1) and (5);
        if Edge cache is full then
            $j = argmin\{\hat{p}_k(t)\tilde{\gamma}_k(t) :$
              $d_k$ is currently stored in the edge cache$\}$;
            if $\hat{p}_i(t)\tilde{\gamma}_i(t) > \hat{p}_j(t)\tilde{\gamma}_j(t)$ then
                Load $d_i$ into the cache and evict $d_j$;
        else
            Load $d_i$ into the cache;
        end
end
```

B. Regret Upper Bound and Asymptotic Optimality

We provide theoretical performance guarantees for the proposed KL-LCB based edge caching policy by deriving a regret upper bound in Theorem 2.

Theorem 2. *For the proposed KL-LCB based policy, we have*

$$\limsup_{n\to+\infty}\frac{Regret(n)}{\log n} \le \sum_{i\in\mathcal{S}}\frac{p_i\gamma_i - p_{K+1}\gamma_{K+1}}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1}-p_i(c_1-c_0)}{p_i(c_2-c_1)}\right)\cdot p_i},$$

where $\mathcal{S} = \{1 \le i \le K : p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}\}$.

The upper bound derived in Theorem 2 matches the lower bound derived in Theorem 1. Therefore, we can conclude that the proposed KL-LCB based edge caching policy is *asymptotically optimal* if the set $\mathcal{S}$ is not empty.

It is well known that the KL-UCB policy achieves asymptotically optimal regrets for conventional stochastic MAB problems with Bernoulli-distributed rewards. One question is whether the KL-LCB based edge caching policy proposed in this paper is a simple application of the KL-UCB policy. The answer is no. Although the miss cost estimation in the proposed edge caching policy is symmetric to the reward estimation in KL-UCB, the theoretical analysis of the proposed edge caching policy is much more challenging for the following reasons:

- The nature of edge caching systems imposes significant complexity into performance analysis. In particular, the data items that can be cached at a time slot t depend on the data items that were already cached at the previous time slot $t-1$, while in the original KL-UCB setting, the action space have no such time correlations.
- The data request process introduces new dynamics compared to the original MAB settings. Specifically, the action space at each slot depend on a random data request, while in the original KL-UCB setting, there is no such dependency.

Simply applying existing analysis of the KL-UCB policy cannot resolve these challenges. Instead, we need to develop novel approaches in this paper to analyze the theoretical performance. In particular,

- We characterize the nature of the caching systems by exploring the relationship between $T_i^{\text{miss}}(n)$, $T_i^{\text{out}}(n)$ and $T_i^{\text{in}}(n)$, $1 \le i \le N$.
- Instead of showing the convergence in expectation for critical statistics as in the conventional MAB analysis, our setting requires much stronger almost-sure convergence results, which need non-trivial new analysis.

C. Sketch of Proof for Theorem 2

In this section, we briefly introduce the proof for Theorem 2. First, we need to establish a few lemmas.

Lemma 1. *The regret of the KL-LCB based edge caching policy can be upper bounded as* $Regret(n) \le \sum_{i=1}^{K}\mathbb{E}[T_i^{out}(n)]p_i\gamma_i - \mathbb{E}[T_{K+1}^{in}(n)]p_{K+1}\gamma_{K+1}$.

Lemma 1 provides a regret upper bound related to the costs introduced by the optimal choice set (i.e., $d_i, 1 \le i \le K$) and best suboptimal choice (i.e., d_{K+1}). The proof of this lemma leverages the cache capacity constraint, and is presented in Appendix B. To connect this upper bound with Theorem 2, we need to characterize $\mathbb{E}[T_i^{\text{out}}(n)]$, $1 \le i \le K$, and $\mathbb{E}[T_{K+1}^{\text{in}}(n)]$ under the proposed KL-LCB based policy.

Lemma 2. *Under the KL-LCB based policy, for* $1 \le i \le K$, *we have, if* $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$, *then*

$$\limsup_{n\to+\infty}\frac{\mathbb{E}\left[T_i^{out}(n)\right]}{\log n} \le 1\Big/\left(p_i\cdot D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1}-p_i(c_1-c_0)}{p_i(c_2-c_1)}\right)\right), \quad (6)$$

otherwise $\lim_{n\to+\infty}\mathbb{E}\left[T_i^{out}(n)\right]/\log n = 0$.

Lemma 2 shows an upper bound for $\mathbb{E}[T_i^{\text{out}}(n)]$, $1 \le i \le K$, under the proposed KL-LCB based policy. The proof of Lemma 2 is shown in Appendix C. Next, we will show a relationship between $\mathbb{E}[T_i^{\text{out}}(n)]$, $1 \le i \le K$, and $\mathbb{E}[T_{K+1}^{\text{in}}(n)]$, which is the most critical and challenging part for the proof of Theorem 2.

Lemma 3. *For* $1 \le i \le K$, *if* $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$, *then under the KL-LCB based edge caching policy, we have*

$$\lim_{n\to+\infty}\frac{\mathbb{E}[T_{K+1}^{in}(n)]}{\sum_{i=1}^{K}\mathbb{E}[T_i^{out}(n)]} = 1.$$

The proof of this lemma is presented in Appendix D. Lemma 3 indicates that under the KL-LCB based edge caching policy, the duration of time when the cache content is not the optimal choice set (i.e., $\sum_{i=1}^{K}\mathbb{E}[T_i^{\text{out}}(n)]$) is approximately equal to the duration of time when the best suboptimal choice is stored in the cache (i.e, $\mathbb{E}[T_{K+1}^{\text{in}}(n)]$). In other words, when the time horizon n is large, the proposed policy almost only gets confused with the best suboptimal data item d_{K+1}, and is confident that other suboptimal data items should not be cached. Intuitively, this result makes a lot of sense, because other suboptimal data items have a larger gap with the optimal choice set and should be easier to distinguish. However, proving this lemma rigorously requires establishing the almost-sure convergence results for critical statistics such as $\tilde{q}_i(t)$, $T_i^{\text{miss}}(t)$, etc, and is highly non-trivial.

Now, we are ready to prove Theorem 2.

Proof of Theorem 2. Combining Lemmas 1, 2 and 3, we have

$$\begin{aligned}
&\limsup_{n\to+\infty}\frac{Regret(n)}{\log n}\\
&\overset{(a)}{\le}\limsup_{n\to+\infty}\frac{\sum_{i=1}^{K}\mathbb{E}[T_i^{\text{out}}(n)]p_i\gamma_i - \mathbb{E}[T_{K+1}^{\text{in}}(n)]p_{K+1}\gamma_{K+1}}{\log n}\\
&\overset{(b)}{=}\limsup_{n\to+\infty}\frac{\sum_{i=1}^{K}\mathbb{E}[T_i^{\text{out}}(n)](p_i\gamma_i - p_{K+1}\gamma_{K+1})}{\log n}\\
&\overset{(c)}{\le}\sum_{i\in\mathcal{S}}\frac{p_i\gamma_i - p_{K+1}\gamma_{K+1}}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1}-p_i(c_1-c_0)}{p_i(c_2-c_1)}\right)\cdot p_i},
\end{aligned}$$

where $\mathcal{S} = \{1 \le i \le K : p_i(c_1-c_0) < p_{K+1}\gamma_{K+1}\}$. Note that steps (a), (b) and (c) leverage Lemmas 1, 3 and 2, respectively. □

VI. Discussion and Generalization

In the section, we extend our design to allow for non-identical data sizes, discuss the benefit of additional observations and explain why the regret bounds do not scale up with $N-K$.

A. Non-identical Data Sizes

Although our design has been under the assumption of unit data sizes, we can generalize the proposed KL-LCB policy for non-identical data sizes. Let s_i denote the size of the data item d_i, $1 \leq i \leq N$. When p_i and q_i are known, the service cost minimization problem can be cast into a knapsack problem, which is non-trivial to solve. However, when the cache size is large, storing the data items with the largest $p_i\gamma_i/s_i$ values in the edge cache is a good approximation for the optimal solution. This idea has been adopted by a few caching policies and system designs [2], [6]. Similarly, to allow for non-identical data sizes, we could generalize the KL-LCB based policy by replacing all $\hat{p}_i(t)\tilde{\gamma}_i(t)$ in Algorithm 2 with $\hat{p}_i(t)\tilde{\gamma}_i(t)/s_i$. The regret analysis for this generalized policy becomes more challenging due to the approximation algorithm for the knapsack problem and is beyond the scope of this paper. Instead, we compare the performance of this extended policy with other benchmarks through numerical simulations in Section VII, Experiment 2.

B. Additional Observation Opportunities

In this paper, we assume that whether a data item d_i can be accessed from the intermediate cache is unknown, and we can obtain an observation only when the request of d_i is not fulfilled at the edge and redirected to the intermediate cache. The unknown parameter q_i is then estimated based on such observations. Notably, we could benefit from additional observation opportunities by enabling the intermediate cache to send the indices of its content to the edge cache at some cost c_{share}. And, at each time slot, the edge cache could decide whether to query the content indices in the intermediate cache and pay the additional cost. The optimal index sharing decision could depend on the value of c_{share} and the amount of observations that have been collected so far. At the early stage, index sharing is plausible to facilitate exploration. Later on, the index sharing may not be preferred to avoid unnecessary costs. The additional observations have been investigated for conventional stochastic MAB problems [32], [33]. However, the optimal decision in caching applications still remains unknown due to the additional caching constraints discussed in Section III-C and deserves further explorations.

C. Why does the regret not scale up with $N-K$?

For the conventional stochastic combinatorial semi bandit problem with linear reward functions, it is shown in [34] that the regret lower bound scales up with $K(N-K)$. However, the regret bounds derived in Theorems 1 and 2 only scale up with K, and are independent of $N-K$. This is due to a special structure of the caching problem, i.e., the popularity distribution always satisfies $\sum_{i=1}^{N} p_i = 1$ as N scales up. Combining this with the fact that $c_0 < c_1 < c_2$, the expected service cost incurred in each round is always bounded by c_2, regardless of N or $N-K$. Instead, for conventional stochastic combinatorial semi bandit with linear reward functions, the cost or reward in each round will scale up with the number of arms that can be played (i.e., $N-K$), which is a key property used to prove the scaling factor in the lower bound [34], [35].

VII. Numerical Evaluation

In this section, we present the numerical simulations. In Experiment 1 and 2, we compare the proposed policies with a few benchmarks for identical and non-identical data sizes, respectively. In Experiment 3, we evaluate the regret of the KL-LCB policy under different N values.

Caching policies: We evaluate the performance of the following policies that can be categorized into two classes.

Class 1: Policies that do not consider service costs

- Opt-Hit: The optimal policy that maximizes hit ratios in an *idealized* scenario where p_i's are known. It will cache the items with the largest p_i if data sizes are identical.
- LFU: Cache the items with the largest request frequency.
- LRU: Cache the most recently used items.

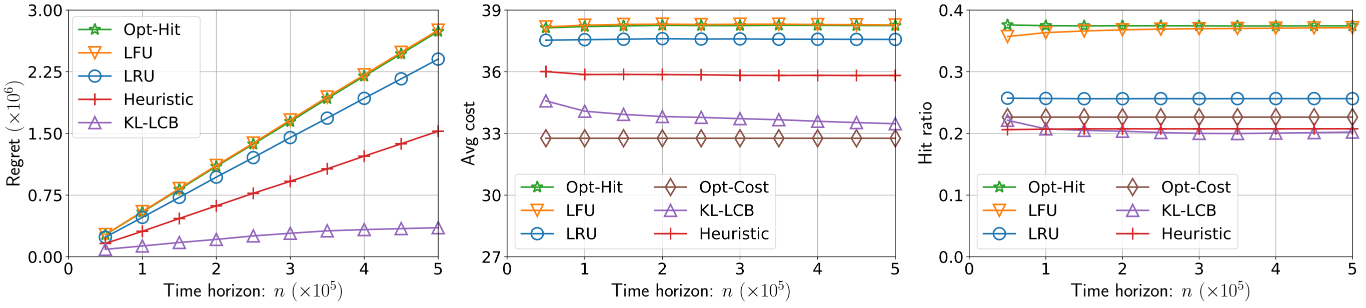
Class 2: Policies that take account of service costs

- Opt-Cost: The optimal policy that minimizes the accumulated service cost in an *idealized* scenario where p_i and q_i are known, i.e., the policy π^* introduced in Section II-B.
- Heuristic: The policy proposed in Algorithm 1.
- KL-LCB: The policy proposed in Algorithm 2.

Experiment 1: Consider a total number of 1000 data items. Set the cache capacity $K = 200$, $c_0 = 1$, $c_1 = 5$, $c_2 = 100$. Assume that data popularities follow a Zipf's distribution, which has been validated by real data traces [36], [37], [1]. Specifically, set $p_i = b \cdot i^{-0.4}$, $1 \leq i \leq 1000$, where $b = 1/\sum_{i=1}^{1000} i^{-0.4} = 9.61 \times 10^{-3}$ is the normalization factor. Set $q_i = 0.2$ for $1 \leq i \leq 500$ and $q_i = 0.9$ for $501 \leq i \leq 1000$. Thus, a more popular data item will have a larger probability (i.e., $1-q_i$) to be stored in the intermediate cache. Notably, in the main paper, we assumed that data items are indexed such that $p_i\gamma_i$ is decreasing for the ease of presentation. However, in this experiment, the data items are indexed such that the popularity p_i is decreasing.

In Fig. 2a, we plot the regrets, i.e., the differences in accumulated costs between a policy and the optimal policy Opt-Cost. The KL-LCB based policy achieves a sublinear regret which is significantly smaller than all the alternatives. Although the heuristic policy takes account of the service costs, its regret still increases linearly like other benchmarks that do not consider service costs, due to the reason discussed in Section III-A. Moreover, the Opt-Hit policy achieves considerably worse performance than KL-LCB, which implies that other popularity-driven polices not simulated in this experiment may also have a large performance gap with KL-LCB, if their goal is maximizing hit ratios.

In Fig. 2b, we illustrate the average cost to serve a data request. The proposed KL-LCB based policy converges to the Opt-Cost policy by strategically learning unknown p_i and q_i, while all the other benchmarks are trapped in suboptimal

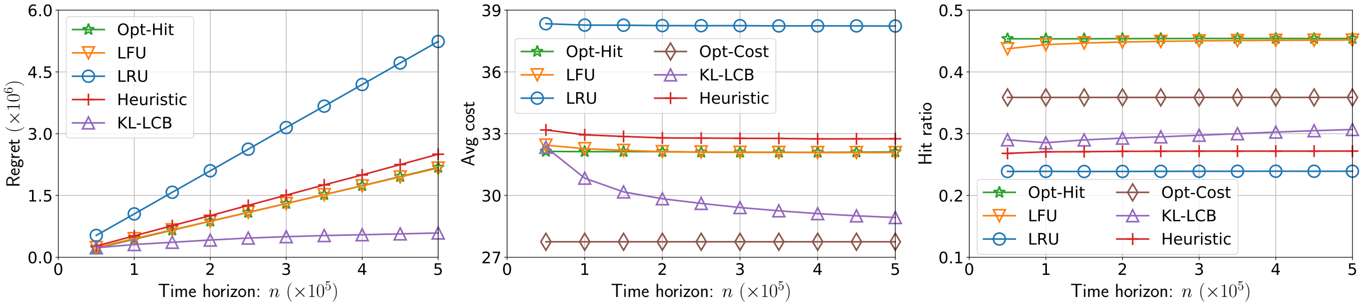


(a)

(b)

(c)

Fig. 2: Performance evaluation with identical data sizes.



(a)

(b)

(c)

Fig. 3: Performance evaluation with non-identical data sizes.

solutions. Furthermore, it can be verified in Fig. 2c that traditional popularity-driven policies, including Opt-Hit, LFU and LRU, indeed achieve high hit ratios, but higher hit ratios cannot guarantee lower service costs. Therefore, the carefully designed cost learning procedure proposed in the KL-LCB policy is critical for achieving good performance.

Experiment 2: In Section VI, we generalize the KL-LCB policy to allow for non-identical data sizes. In this experiment, we compare this generalized policy with pre-mentioned benchmarks that are also extended for non-identical data sizes. Using a similar approximation for the optimal solution of the knapsack problem introduced in Section VI, we let the Opt-Hit policy caches the items with the largest p_i/s_i values. We extend the LFU policy by caching the items with the largest $\hat{p}_i(t)/s_i$, which is an approximation of Hit-Opt when popularities are unknown. For the LRU policies, multiple least recently used items could be evicted to make room for the newly requested one. The Opt-Cost policy knows p_i and q_i parameters and stores the items with the largest $p_i\gamma_i/s_i$ values. The heuristic policy proposed in Algorithm 1 could be extended by replacing all $\hat{p}_i(t)$ with $\hat{p}_i(t)/s_i$. The regret of a policy π is the cost differences between π and Opt-Cost.

We set the cache size $K = 500$ and select the data size s_i from a uniform distribution between 1 and 10. All the other settings are identical to those in Experiment 1. The numerical results for regrets, averages costs and hit ratios are presented in Fig. 3. Similar to the results in Experiment 1, the extended KL-LCB policy achieves a sublinear regret which is much better than the other benchmarks (Fig. 3a). And the average cost of KL-LCB converges to Opt-Cost (Fig. 3b)

Experiment 3: Theorems 1 and 2 show that the scaling factor of the regret only depends on the p_i and q_i values of the first $K+1$ items and does not scale up with the total number of items N. In this experiment, we evaluate the regret of the proposed KL-LCB policy with different N values. To verify the theoretical results, we would like to construct p_i and q_i, such that their values do not change as N scales up for $1 \leq i \leq K+1$. In particular, set the cache capacity $K = 100$, $c_0 = 1$, $c_1 = 5$, $c_2 = 100$. We let $p_i = b_1 \cdot i^{-0.4}$ for $1 \leq i \leq K+1$, and $p_i = b_2 \cdot i^{-0.4}$ for $K+2 \leq i \leq N$, where b_1 and b_2 are choosing such that $\sum_{i=1}^{K+1} p_i = \sum_{i=K+2}^{N} p_i = 0.5$. Set $q_i = 0.5$ for $1 \leq i \leq K+1$ and $q_i = 0.2$ for $K+2 \leq i \leq N$. We simulate the regret for $N = 1000, 5000, 10000$, respectively and present the results in Fig. 4. It can be observed that the regret does not scale up with N.

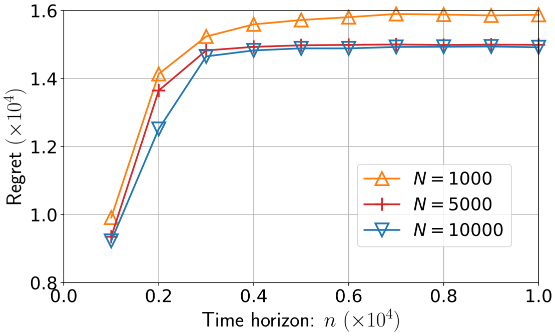


Fig. 4: Regret of KL-LCB policy with different N values.

VIII. Conclusion

Existing cost-based caching policies typically assume known and fixed costs to handle cache misses, which is not always the case in real systems. Various sources of uncertainties exist in the process of fetching missed data from the backend in a content delivery network, including communication failures, packet drops, dynamically changing cache contents, etc. To address this issue, we focused on an edge caching scenario where the miss costs are random with unknown statistics and developed novel caching policies that learn the unknown statistics efficiently. By presenting a carefully-designed example, we first show that a heuristic learning design could induce significant caching inefficiency. We then derived a regret lower bound for any uniformly good policy. Inspired by the "optimism in the face of uncertainty" principle in online learning literature, we developed a KL-LCB based edge caching policy and proved that it achieves the regret lower bound and is asymptotically optimal. Extensive numerical experiments indicate the proposed policy significantly improves caching performance over other benchmarks. The novel techniques used in this work to handle caching constraints and dynamics could be potentially leveraged to design and analyze learning mechanisms for other systems.

Appendix A
Proof of Theorem 1

As discussed in Section II-B, the optimal policy with known p_i and q_i always store the data items d_i, $1 \le i \le K$ in the cache. The regret can be bounded as

$$\begin{aligned} Regret(n) &= \sum_{i=1}^{K} \mathbb{E}[T_i^{\text{out}}(n)] p_i\gamma_i - \sum_{i=K+1}^{N} \mathbb{E}[T_i^{\text{in}}(n)] p_i\gamma_i \\ &\ge \sum_{i=1}^{K} \mathbb{E}[T_i^{\text{out}}(n)] p_i\gamma_i - \sum_{i=K+1}^{N} \mathbb{E}[T_i^{\text{in}}(n)] p_{K+1}\gamma_{K+1} \\ &\ge \sum_{i=1}^{K} \mathbb{E}[T_i^{\text{out}}(n)] (p_i\gamma_i - p_{K+1}\gamma_{K+1}), \end{aligned} \tag{7}$$

where the last inequality is due to $\sum_{i=K+1}^{N} \mathbb{E}[T_i^{\text{in}}(n)] \le \sum_{i=1}^{K} \mathbb{E}[T_i^{\text{out}}(n)]$. The key step to prove this theorem is deriving a lower bound for $\mathbb{E}[T_i^{\text{out}}(n)]$ or $\mathbb{E}[T_i^{\text{miss}}(n)]$, $1 \le i \le K$.

Without loss of generality, we focus on $\mathbb{E}[T_i^{\text{miss}}(n)]$ for $i = 1$. To analyze the lower bound for $\mathbb{E}[T_1^{\text{miss}}(n)]$, we introduce two instances ν and ν', where ν is the original instance of our problem and ν' is the instance that could confuse the policy. In particular, under ν', we set $p'_i = p_i$ for $1 \le i \le N$ and $q'_i = q_i$ for $2 \le i \le N$. Assume that $p_1(c_1 - c_0) < p_{K+1}\gamma_{K+1}$. For $\epsilon > 0$, we can select $q'_1 \in (0, q_1)$ such that

$$\begin{aligned} D_{KL}\left(q_1, \frac{p_{K+1}\gamma_{K+1} - p_1(c_1 - c_0)}{p_1(c_2 - c_1)}\right) &< D_{KL}(q_1, q'_1) \\ < D_{KL}\left(q_1, \frac{p_{K+1}\gamma_{K+1} - p_1(c_1 - c_0)}{p_1(c_2 - c_1)}\right) &+ \epsilon. \end{aligned} \tag{8}$$

Define $\gamma'_i = q'_i c_2 + (1 - q'_i)c_1 - c_0$. Under the setting of ν', d_1 becomes a suboptimal data item since $p'_1\gamma'_1 < p'_i\gamma'_i$ for $2 \le i \le K+1$. The optimal solution is to store items d_i, $2 \le i \le K+1$, in the cache.

Define an event $A = \{T_1^{\text{out}}(n) \ge n/2\}$ and its complement $A^c = \{T_1^{\text{out}}(n) < n/2\}$. Let $Regret'(n)$ denote the regret achieved under the predefined instance ν'. We have

$$\begin{aligned} Regret(n) &\ge \mathbb{P}_\nu[A](p_1\gamma_1 - p_{K+1}\gamma_{K+1})n/2, \\ Regret'(n) &\ge \mathbb{P}_{\nu'}[A^c](p_{K+1}\gamma_{K+1} - p_1\gamma'_1)n/2, \end{aligned}$$

which yields

$$\begin{aligned} &Regret(n) + Regret'(n) \\ &\ge (\mathbb{P}_\nu[A] + \mathbb{P}_{\nu'}[A^c]) \\ &\quad \cdot \min\left\{p_1 q_1 - p_{K+1}\gamma_{K+1}, p_{K+1}q_{K+1} - p_1\gamma'_1\right\} n/2 \\ &\ge \exp\left(-\mathcal{D}(\mathbb{P}_\nu, \mathbb{P}_{\nu'})\right) \\ &\quad \cdot \min\left\{p_1\gamma_1 - p_{K+1}\gamma_{K+1}, p_{K+1}\gamma_{K+1} - p_1\gamma'_1\right\} n/4. \end{aligned} \tag{9}$$

The last inequality holds because of the Bretagnolle-Huber inequality and $\mathcal{D}(\mathbb{P}_\nu, \mathbb{P}_{\nu'})$ is the relative entropy between the distributions $\mathbb{P}_\nu$ and $\mathbb{P}_{\nu'}$. Next, we will derive the expression of $\mathcal{D}(\mathbb{P}_\nu, \mathbb{P}_{\nu'})$.

Let R_t, A_t, C_t denote the data item requested at t, the set of data items that are stored in the cache after serving R_t based on the caching policy, and the cost to serve R_t, respectively. R_t, A_t, C_t are random variables. Let A_0 denote the set of data items that are initially stored in the cache. We use R_i^j, $i < j$, to denote the list of random variables $R_i, R_{i+1}, \cdots, R_j$. Similarly, we can define A_i^j, C_i^j.

Given the time horizon n (i.e, $1 \le t \le n$), we have

$$\mathcal{D}(\mathbb{P}_\nu, \mathbb{P}_{\nu'}) = \mathbb{E}_\nu\Big[\log \frac{\mathbb{P}_\nu}{\mathbb{P}_{\nu'}}[R_1^n, A_1^n, C_1^n \mid A_0]\Big], \tag{10}$$

Under both instances, we have

$$\begin{aligned} &\mathbb{P}\left[R_1^n, A_1^n, C_1^n \mid A_0\right] \\ &= \mathbb{P}[R_1, A_1, C_1 \mid A_0] \cdot \prod_{t=2}^{n} \mathbb{P}\left[R_t, A_t, C_t \mid R_1^{t-1}, A_0^{t-1}, C_1^{t-1}\right] \\ &= \mathbb{P}[R_1, A_1 \mid A_0] \cdot \mathbb{P}[C_1 \mid R_1, A_1, A_0] \\ &\quad \cdot \prod_{t=2}^{n} \mathbb{P}\left[R_t, A_t \mid R_1^{t-1}, A_0^{t-1}, C_1^{t-1}\right] \mathbb{P}\left[C_t \mid R_1^t, A_0^t, C_1^{t-1}\right] \\ &= \mathbb{P}[R_1, A_1 \mid A_0] \cdot \mathbb{P}[C_1 \mid R_1, A_0] \\ &\quad \cdot \prod_{t=2}^{n} \mathbb{P}\left[R_t, A_t \mid R_1^{t-1}, A_0^{t-1}, C_1^{t-1}\right] \mathbb{P}\left[C_t \mid R_t, A_{t-1}\right], \end{aligned}$$

where the last equality is due to the fact that the cost C_t only depends on the request R_t and the data items stored in the cache before the R_t arrives, i.e., A_{t-1}. Moreover, we have

$$\begin{aligned} &\frac{\mathbb{P}_\nu}{\mathbb{P}_{\nu'}}\left[R_1, A_1 \mid A_0\right] = 1, \\ &\frac{\mathbb{P}_\nu}{\mathbb{P}_{\nu'}}\left[R_t, A_t \mid R_1^{t-1}, A_0^{t-1}, C_1^{t-1}\right] = 1, \quad 2 \le t \le n, \end{aligned}$$

since under the two instances, the popularity distributions are identical and the policy will make the same decision A_t given the same history $R_1^{t-1}, A_0^{t-1}, C_1^{t-1}$.

Therefore, we have

$$\frac{\mathbb{P}_\nu}{\mathbb{P}_{\nu'}}\left[R_1^n, A_1^n, C_1^n \mid A_0\right] = \prod_{t=1}^{n} \frac{\mathbb{P}_\nu}{\mathbb{P}_{\nu'}}\left[C_t \mid R_t, A_{t-1}\right].$$

For each time slot t, define the event $\mathcal{E}_t$ as

$$\mathcal{E}_t = \{R_t = d_1, d_1 \notin A_{t-1}\}.$$

Let $\mathcal{E}_t^c$ denote the complementary event of $\mathcal{E}_t$. We have

$$\frac{\mathbb{P}_\nu}{\mathbb{P}_{\nu'}}\left[C_t \mid R_t, A_{t-1}, \mathcal{E}_t^c\right] = 1,$$

because the two instances only differs in the first data item. The relative entropy $\mathcal{D}(\mathbb{P}_\nu, \mathbb{P}_{\nu'})$ defined in (10) can be simplified as

$$\begin{aligned}
\mathcal{D}(\mathbb{P}_\nu, \mathbb{P}_{\nu'}) &= \sum_{t=1}^{n} \mathbb{E}_\nu\left[\log\left(\frac{\mathbb{P}_\nu}{\mathbb{P}_{\nu'}}\left[C_t \mid R_t, A_{t-1}\right]\right)\right] \\
&= \sum_{t=1}^{n} \mathbb{E}_\nu\left[\mathbf{1}(\mathcal{E}_t)\log\left(\frac{\mathbb{P}_\nu}{\mathbb{P}_{\nu'}}\left[C_t \mid R_t, A_{t-1}\right]\right)\right. \\
&\qquad\qquad \left. + \mathbf{1}(\mathcal{E}_t^c)\log\left(\frac{\mathbb{P}_\nu}{\mathbb{P}_{\nu'}}\left[C_t \mid R_t, A_{t-1}\right]\right)\right] \\
&= \sum_{t=1}^{n} \mathbb{E}_\nu\left[\mathbf{1}(\mathcal{E}_t)\log\left(\frac{\mathbb{P}_\nu}{\mathbb{P}_{\nu'}}\left[C_t \mid R_t, A_{t-1}\right]\right)\right] \\
&= \sum_{t=1}^{n} \mathbb{E}_\nu\left[\mathbf{1}(\mathcal{E}_t) \cdot D_{KL}(q_1, q_1')\right] \\
&= \mathbb{E}_\nu[T_i^{\text{miss}}(n)] \cdot D_{KL}(q_1, q_1'). \qquad (11)
\end{aligned}$$

Combining (8), (9) and (11), we have

$$\begin{aligned}
&Regret(n) + Regret'(n) \\
&\geq \exp\left(-\mathbb{E}_\nu[T_i^{\text{miss}}(n)] \cdot D_{KL}(q_1, q_1')\right) \\
&\quad \cdot \min\left\{p_1\gamma_1 - p_{K+1}\gamma_{K+1}, p_{K+1}\gamma_{K+1} - p_1\gamma_1'\right\}\frac{n}{4} \\
&\geq \exp\Bigg(- \mathbb{E}_\nu[T_i^{\text{miss}}(n)] \\
&\qquad \cdot \left(D_{KL}\left(q_1, \frac{p_{K+1}\gamma_{K+1} - p_1(c_1 - c_0)}{p_1(c_2 - c_1)}\right) + \epsilon\right)\Bigg) \\
&\quad \cdot \min\left\{p_1\gamma_1 - p_{K+1}\gamma_{K+1}, p_{K+1}\gamma_{K+1} - p_1\gamma_1'\right\}\frac{n}{4},
\end{aligned}$$

which indicates that, for $\forall\epsilon > 0$, we have

$$\begin{aligned}
\mathbb{E}_\nu[T_i^{\text{miss}}(n)] &\geq \frac{1}{D_{KL}\left(q_1, \frac{p_{K+1}\gamma_{K+1} - p_1(c_1 - c_0)}{p_1(c_2 - c_1)}\right) + \epsilon} \\
&\quad \cdot \log\frac{n \cdot \min\left\{p_1\gamma_1 - p_{K+1}\gamma_{K+1}, p_{K+1}\gamma_{K+1} - p_1\gamma_1'\right\}}{Regret(n) + Regret'(n)}.
\end{aligned}$$

Recall that the uniformly good policy has $Regret(n) = o(n^\alpha)$ and $Regret'(n) = o(n^\alpha)$ for $\forall\alpha > 0$. We have

$$\begin{aligned}
\liminf_{n\to+\infty}\frac{\mathbb{E}_\nu[T_1^{\text{miss}}(n)]}{\log n} &\geq \frac{1}{D_{KL}\left(q_1, \frac{p_{K+1}\gamma_{K+1} - p_1(c_1 - c_0)}{p_1(c_2 - c_1)}\right)} \\
&\quad \cdot \liminf_{n\to+\infty}\Bigg(1 - \frac{\log(Regret(n) + Regret'(n))}{\log n} \\
&\quad + \frac{\log\min\left\{p_1\gamma_1 - p_{K+1}\gamma_{K+1}, p_{K+1}\gamma_{K+1} - p_1\gamma_1'\right\}}{\log n}\Bigg) \\
&\geq \frac{1}{D_{KL}\left(q_1, \frac{p_{K+1}\gamma_{K+1} - p_1(c_1 - c_0)}{p_1(c_2 - c_1)}\right)}.
\end{aligned}$$

Using a similar approach, we can prove that for any $1 \leq i \leq K$, if $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$, then

$$\begin{aligned}
&\liminf_{n\to+\infty}\frac{\mathbb{E}_\nu[T_i^{\text{miss}}(n)]}{\log n} \\
&\qquad \geq 1\Big/D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_1(c_2 - c_1)}\right). \qquad (12)
\end{aligned}$$

Combining (7) and (12) and using the weak law of large numbers, we have

$$\begin{aligned}
&\liminf_{n\to+\infty}\frac{Regret(n)}{\log n} \\
&\qquad \geq \liminf_{n\to+\infty}\sum_{i=1}^{K}\frac{\mathbb{E}[T_i^{\text{out}}(n)]}{\log n}(p_i\gamma_i - p_{K+1}\gamma_{K+1}) \\
&\qquad = \liminf_{n\to+\infty}\sum_{i=1}^{K}\frac{\mathbb{E}[T_i^{\text{miss}}(n)]/p_i}{\log n}(p_i\gamma_i - p_{K+1}\gamma_{K+1}) \\
&\qquad \geq \sum_{i\in\mathcal{S}}\frac{p_i\gamma_i - p_{K+1}\gamma_{K+1}}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_1(c_2 - c_1)}\right)p_i},
\end{aligned}$$

where $\mathcal{S} = \{1 \leq i \leq K : p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}\}$.

Appendix B
Proof of Lemma 1

Based on the definition of the regret, we have

$$\begin{aligned}
Regret(n) &= \sum_{i=1}^{N}\mathbb{E}[T_i^{\text{out}}(n)]p_i\gamma_i - \sum_{i=K+1}^{N} np_i\gamma_i \\
&= \sum_{i=1}^{K}\mathbb{E}[T_i^{\text{out}}(n)]p_i\gamma_i - \sum_{i=K+1}^{N}\mathbb{E}[T_i^{\text{in}}(n)]p_i\gamma_i \\
&\leq \sum_{i=1}^{K}\mathbb{E}[T_i^{\text{out}}(n)]p_i\gamma_i - \mathbb{E}[T_{K+1}^{\text{in}}(n)]p_{K+1}\gamma_{K+1}.
\end{aligned}$$

Appendix C
Proof of Lemma 2

Proof of Lemma 2. We will first derive bounds for $\mathbb{E}[T_i^{\text{miss}}(n)]$. Define $\underline{D}_{KL}(p,q) = D_{KL}(p,q) \cdot \mathbf{1}(p \geq q)$. Define $\hat{q}_i^{(s)} = \hat{q}_i(t)$ with s samples (i.e., $T_i^{\text{miss}}(t) = s$). For $\epsilon > 0$, define

$$\begin{aligned}
\tau_i = \min\Big\{&t \geq 1 : \\
&\max_{1\leq s\leq n}\{\underline{D}_{KL}(\hat{q}_i^{(s)}, q_i + \epsilon) - \log f(t)/s\} \leq 0\Big\}.
\end{aligned}$$

Thus, for any $t \geq \tau_i$, we must have $\tilde{q}_i(t) \leq q_i + \epsilon$ regardless of the value of $T_i^{\text{miss}}(t)$.

We can bound $\mathbb{P}[\tau_i > t]$ as

$$\begin{aligned}
&\mathbb{P}[\tau_i > t] \leq \mathbb{P}\left[\exists s \in [1,n] : \underline{D}_{KL}(\hat{q}_i^{(s)}, q_i + \epsilon) > \log f(t)/s\right] \\
&\leq \sum_{s=1}^{n}\mathbb{P}\left[\underline{D}_{KL}(\hat{q}_i^{(s)}, q_i + \epsilon) > \log f(t)/s\right] \\
&= \sum_{s=1}^{n}\mathbb{P}\left[D_{KL}(\hat{q}_i^{(s)}, q_i + \epsilon) > \log f(t)/s,\ \hat{q}_i^{(s)} \geq q_i + \epsilon\right]
\end{aligned}$$

$$\le \sum_{s=1}^{n} \mathbb{P}\left[D_{KL}(\hat{q}_i^{(s)}, q_i) - 2\epsilon^2 > \log f(t)/s,\ \hat{q}_i^{(s)} \ge q_i + \epsilon\right]$$
$$\le \sum_{s=1}^{n} \exp\left(-s(2\epsilon^2 + \log f(t)/s)\right) \le \frac{1}{2\epsilon^2 f(t)}.$$

Let $\tau = \max_{K+1\le i\le N} \tau_i$. We have

$$\mathbb{P}[\tau > t] = \mathbb{P}\Big[\max_{K+1\le i\le N}\{\tau_i\} > t\Big] \le \sum_{i=K+1}^{N} \mathbb{P}[\tau_i > t].$$

Therefore, we have

$$\begin{aligned}\mathbb{E}[\tau] &= \sum_{t=0}^{+\infty} \mathbb{P}[\tau > t] \le \sum_{t=0}^{+\infty}\sum_{i=K+1}^{N} \mathbb{P}[\tau_i > t] \\ &\le \sum_{t=0}^{+\infty}\sum_{i=K+1}^{N} \frac{1}{2\epsilon^2 f(t)} = \frac{N-K}{2\epsilon^2}\sum_{t=0}^{+\infty}\frac{1}{f(t)} \\ &\le \frac{N-K}{2\epsilon^2}\int_{t=0}^{+\infty}\frac{1}{f(t)}dt \le \frac{2(N-K)}{\epsilon^2}. \qquad (13)\end{aligned}$$

Define $\tau_i' = \min\{t \ge 1 : \hat{p}_i(s) \in (p_i - \epsilon, p_i + \epsilon) \text{ for all } s \ge t\}$. Let $\tau' = \max_{1\le i\le N} \tau_i'$, we have

$$\begin{aligned}\mathbb{P}[\tau' \ge t] &\le \sum_{i=1}^{N} \mathbb{P}[\tau_i' \ge t] \le \sum_{i=1}^{N}\sum_{s=t}^{+\infty} \mathbb{P}[\hat{p}_i(s) \notin (p_i - \epsilon, p_i + \epsilon)] \\ &\le \sum_{i=1}^{N}\sum_{s=t}^{+\infty} \exp(-2\epsilon^2 s) \le \frac{N}{2\epsilon^2}\exp(-2\epsilon^2(t-1)),\end{aligned}$$

which yields

$$\begin{aligned}\mathbb{E}[\tau'] &= \sum_{t=0}^{+\infty} \mathbb{P}[\tau' > t] \le \sum_{i=1}^{+\infty} \frac{N}{2\epsilon^2}\exp(-2\epsilon^2(t-1)) \\ &\le \frac{N\exp(2\epsilon^2)}{4\epsilon^4}. \qquad (14)\end{aligned}$$

For $1 \le i \le K$, $\mathbb{E}[T_i^{\text{miss}}(n)]$ can be upper bounded as

$$\begin{aligned}&\mathbb{E}[T_i^{\text{miss}}(n)] \le \mathbb{E}[\tau] + \mathbb{E}[\tau'] \\ &+ \mathbb{E}\left[\sum_{t=\max\{\tau,\tau'\}}^{n} \mathbf{1}(d_i \text{ is requested and missed at } t)\right]. \quad (15)\end{aligned}$$

Recall that $\tilde{\gamma}_i(t) = \tilde{q}_i(t)c_2 + (1-\tilde{q}_i(t))c_1 - c_0$. Based on the definition of τ and τ', there exists an ϵ' small enough such that $\hat{p}_i\tilde{\gamma}_i(t) \le p_i\gamma_i + \epsilon' \le p_{K+1}\gamma_{K+1} + \epsilon'$ for any $K+1 \le i \le N$ and $t \ge \max\{\tau, \tau'\}$. As a result, we have

$$\begin{aligned}&\{d_i \text{ is requested and missed at } t, 1 \le i \le K, t \ge \max\{\tau,\tau'\}\} \\ &\subseteq \{\hat{p}_i\tilde{\gamma}_i(t) \le p_{K+1}\gamma_{K+1} + \epsilon', 1 \le i \le K, t \ge \max\{\tau,\tau'\}\}.\end{aligned}$$

To derive upper bounds for $\mathbb{E}[T_i^{\text{miss}}(n)]$, $1 \le i \le K$, we first consider the case with $(p_i - \epsilon)(c_1 - c_0) \ge p_{K+1}\gamma_{K+1} + \epsilon'$. For this case with $t \ge \max\{\tau, \tau'\}$, $1 \le i \le K$ and $K+1 \le j \le N$, we have $\hat{p}_i\tilde{\gamma}_i(t) \ge (p_i - \epsilon)(c_1 - c_0) \ge p_{K+1}\gamma_{K+1} + \epsilon' \ge \hat{p}_j\tilde{\gamma}_j(t)$, which indicates that there is no cache miss for $t \ge \max\{\tau, \tau'\}$. Combining it with (13), (14) and (15), we have if $p_i(c_1 - c_0) \ge p_{K+1}(\gamma_{K+1} + \epsilon')$, then

$$\mathbb{E}[T_i^{\text{miss}}(n)] \le \frac{2(N-K)}{\epsilon^2} + \frac{N\exp(2\epsilon^2)}{4\epsilon^4} + 1. \qquad (16)$$

For the case with $(p_i - \epsilon)(c_1 - c_0) < p_{K+1}\gamma_{K+1} + \epsilon'$, define

$$D_{KL}(t) \triangleq D_{KL}\Big(\hat{q}_i(t), \frac{p_{K+1}\gamma_{K+1} + \epsilon' - (p_i - \epsilon)(c_1 - c_0)}{(p_i - \epsilon)(c_2 - c_1)}\Big).$$

Then we have

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=\max\{\tau,\tau'\}}^{n} \mathbf{1}(d_i \text{ is requested and missed at } t)\right] \\ &\le 1 + \mathbb{E}\left[\sum_{t=\max\{\tau,\tau'\}}^{n} \mathbf{1}\Big(d_i \text{ is requested and missed at } t, \right. \\ &\qquad\qquad\qquad \left. D_{KL}(t) \le \frac{\log f(t)}{T_i^{\text{miss}}(t)}\Big)\right] \\ &\le \sum_{s=1}^{n} \mathbb{P}[D_{KL}(s) \le \log f(n)/s] + 1. \qquad (17)\end{aligned}$$

For any δ such that

$$0 < \delta < q_i - \frac{p_{K+1}\gamma_{K+1} + \epsilon' - (p_i - \epsilon)(c_1 - c_0)}{(p_i - \epsilon)(c_2 - c_1)},$$

we have

$$\begin{aligned}&\sum_{s=1}^{n} \mathbb{P}\big[D_{KL}(s) \le \log f(n)/s\big] \\ &\le \sum_{s=1}^{n} \mathbb{P}\Big[\Big\{\hat{q}_i^{(s)} \le q_i - \delta\Big\} \\ &\quad \cup \Big\{D_{KL}\Big(q_i - \delta, \frac{p_{K+1}\gamma_{K+1} + \epsilon' - (p_i - \epsilon)(c_1 - c_0)}{(p_i - \epsilon)(c_2 - c_1)}\Big) \\ &\qquad \le \log f(n)/s\Big\}\Big] \\ &\le \frac{\log f(n)}{D_{KL}\left(q_i - \delta, \frac{p_{K+1}\gamma_{K+1} + \epsilon' - (p_i - \epsilon)(c_1 - c_0)}{(p_i - \epsilon)(c_2 - c_1)}\right)} \\ &\quad + \sum_{s=1}^{n} \mathbb{P}[\hat{q}_i^{(s)} \le q_i - \delta] \\ &\le \frac{\log f(n)}{D_{KL}\left(q_i - \delta, \frac{p_{K+1}\gamma_{K+1} + \epsilon' - (p_i - \epsilon)(c_1 - c_0)}{(p_i - \epsilon)(c_2 - c_1)}\right)} \\ &\quad + \sum_{s=1}^{n} \exp\left(-D_{KL}(q_i - \delta, q_i) \cdot s\right) \\ &\le \frac{\log f(n)}{D_{KL}\left(q_i - \delta, \frac{p_{K+1}\gamma_{K+1} + \epsilon' - (p_i - \epsilon)(c_1 - c_0)}{(p_i - \epsilon)(c_2 - c_1)}\right)} \\ &\quad + 1/D_{KL}(q_i - \delta, q_i) \\ &\le \frac{\log f(n)}{D_{KL}\left(q_i - \delta, \frac{p_{K+1}\gamma_{K+1} + \epsilon' - (p_i - \epsilon)(c_1 - c_0)}{(p_i - \epsilon)(c_2 - c_1)}\right)} + \frac{1}{\delta^2}. \qquad (18)\end{aligned}$$

Combining (15), (17) and (18), we have, for $1 \le i \le K$ if $(p_i - \epsilon)(c_1 - c_0) \ge p_{K+1}\gamma_{K+1} + \epsilon'$, then

$$\begin{aligned}\mathbb{E}\left[T_i^{\text{miss}}(n)\right] &\le \frac{\log f(n)}{D_{KL}\left(q_i - \delta, \frac{p_{K+1}\gamma_{K+1} + \epsilon' - (p_i - \epsilon)(c_1 - c_0)}{(p_i - \epsilon)(c_2 - c_1)}\right)} \\ &\quad + \frac{2(N-K)}{\epsilon^2} + \frac{N\exp(2\epsilon^2)}{4\epsilon^4} + \frac{1}{\delta^2} + 1. \qquad (19)\end{aligned}$$

Note that (16) and (19) hold for any sufficiently small ϵ, ϵ' and δ that are irrelevant to n. Thus, we have, for $1 \le i \le K$, if $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$,

$$\limsup_{n\to+\infty} \mathbb{E}\left[T_i^{\text{miss}}(n)\right] / \log n \le 1 \Big/ D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right) \quad (20)$$

and if $p_i(c_1 - c_0) \ge p_{K+1}\gamma_{K+1}$,

$$\lim_{n\to+\infty} \mathbb{E}\left[T_i^{\text{miss}}(n)\right] / \log n = 0. \quad (21)$$

Moreover, the weak law of large numbers implies that

$$\lim_{n\to+\infty} \mathbb{E}[T_i^{\text{miss}}(n)] / \mathbb{E}[T_i^{\text{out}}(n)] = p_i. \quad (22)$$

Combining (20), (21) and (22) finishes the proof. □

Appendix D
Proof of Lemma 3

In order to prove Lemma 3, we will first introduce some additional lemmas.

Lemma 4. *Under the KL-LCB policy, for any* $1 \le i \le K$ *with* $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$ *and any* $K + 1 \le i \le N$, *we have* $\lim_{t\to+\infty} T_i^{miss}(t) = +\infty$ *and* $\lim_{t\to+\infty} \hat{q}_i(t) = q_i$ *almost surely.*

Proof of Lemma 4. First, we claim that, as $t \to +\infty$ we must have $T_i^{\text{out}}(t) \to +\infty$ almost surely for any $1 \le i \le K$ with $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$ and any $K + 1 \le i \le N$. Suppose towards a contradiction that $\lim_{t\to+\infty} T_i^{\text{out}}(t) < +\infty$. Then there must exist a time τ, such that the data item d_i will always be cached since τ. As a result, we have $\log f(t)/T_i^{\text{miss}}(t) \ge \log f(t)/\tau \to +\infty$ as $t \to +\infty$, which yields $\lim_{t\to+\infty} \tilde{q}_i(t) = 0$ and $\lim_{t\to+\infty} \tilde{\gamma}_i(t) = c_1 - c_0$ according to (4) and (5). Thus, as $t \to +\infty$, we have $\hat{p}_i(t)\tilde{\gamma}_i(t) \to p_i(c_1 - c_0)$ almost surely. Based on the KL-LCB based policy, d_i will be eventually evicted from the cache almost surely, which contradicts the assumption. Therefore, we must have $\lim_{t\to+\infty} T_i^{\text{out}}(t) = +\infty$ almost surely. Furthermore, when d_i is not stored in the edge cache, a miss for d_i happens with a probability p_i independently at each time slot, which indicates that $\lim_{t\to+\infty} T_i^{\text{miss}}(t) = +\infty$ almost surely by the strong law of large numbers.

Recall the definition of $\hat{q}_i(t)$ in (2). For each cache miss, d_i is served from the backend data center with a probability q_i independently. Using the strong law of large numbers, we have $\lim_{t\to+\infty} \hat{q}_i(t) = q_i$ almost surely. □

Lemma 5. *For* $1 \le i \le K$, *if* $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$, *then under the KL-LCB policy, for any small* δ *such that* $0 < \delta < \min\{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0), p_i\gamma_i - p_{K+1}\gamma_{K+1}\}$, *we have*

$$\mathbb{P}\left[\tilde{q}_i(t) \ge \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) + \delta}{p_i(c_2 - c_1)} \textit{ infinitely often}\right] = 0.$$

Proof of Lemma 5. Based on the KL-LCB policy, we have $\tilde{q}_i(t) < \tilde{q}_i(t-1)$, if d_i is not requested at t, or d_i is requested and is a cache hit at t. In other words, we could have $\tilde{q}_i(t) > \tilde{q}_i(t-1)$ only when d_i is requested at t and it is a cache miss. And we will focus on such time stamps to prove the bound.

Without loss of generality, we prove this lemma for $i = 1$. The same approach can be applied to any $1 \le i \le K$. Define events A_t and B_t as

$$A_t = \{d_1 \text{ is requested and missed at } t, \text{ and loaded into the cache after serving the request}\},$$

$$B_t = \left\{\tilde{q}_1(t) \ge \frac{p_{K+1}\gamma_{K+1} - p_1(c_1 - c_0) + \delta}{p_1(c_2 - c_1)}\right\} \cap A_t.$$

We claim that

$$\left\{\left\{\tilde{q}_1(t) \ge \frac{p_{K+1}\gamma_{K+1} - p_1(c_1 - c_0) + \delta}{p_1(c_2 - c_1)}\right\} \text{ infinitely often}\right\} \subseteq \{B_t \text{ infinitely often}\} \quad (23)$$

almost surely, which can be verified by combining Lemma 4, the update rule of the KL-LCB policy and the fact that $\lim_{t\to+\infty} \hat{p}_1(t) = p_1$ almost surely. Next, we will show that B_t infinitely often happens with probability zero.

For a small $\epsilon > 0$, we define events C_t, D_t, E_t as

$$C_t = \{\hat{p}_i\tilde{\gamma}_i(t) \le p_{K+1}\gamma_{K+1} + \delta/2 \text{ for } \forall K + 1 \le i \le N\},$$
$$D_t = \{\hat{q}_1(t) \in (q_1 - \epsilon, q_1 + \epsilon)\},$$
$$E_t = \{\hat{p}_1(t) \in (p_i - \epsilon, p_i + \epsilon)\}.$$

Their complementary events are unlikely to happen. Specifically, we have

$$\mathbb{P}[C_t^c \text{ infinitely often}] = \mathbb{P}[D_t^c \text{ infinitely often}] = \mathbb{P}[E_t^c \text{ infinitely often}] = 0. \quad (24)$$

The equality holds for C_t^c because of Lemma 4, $\tilde{q}_i(t) \le \hat{q}_i(t)$ and $\lim_{t\to+\infty} \hat{p}_i(t) \to p_i$ almost surely. It holds for D_t^c because of Lemma 4 and $\lim_{t\to\infty} \hat{q}_1(t) = q_1$ almost surely. It holds for E_t^c due to the strong law of large numbers.

Let $\sigma_t = \max\{s : d_1 \text{ is evicted at } s, s < t\}$. We have $B_t \subseteq \{B_t \cap C_{\sigma_t} \cap D_{t-1} \cap D_t \cap E_{\sigma_t}\} \cup C_{\sigma_t}^c \cup D_{t-1}^c \cup D_t^c \cup E_{\sigma_t}^c$. In order to show $\mathbb{P}[B_t \text{ infinitely often}] = 0$, it is sufficient to prove $\mathbb{P}[B_t \cap C_{\sigma_t} \cap D_{t-1} \cap D_t \cap E_{\sigma_t} \text{ infinitely often}] = 0$.

Based on the KL-LCB policy, the event $C_{\sigma_t} \cap E_{\sigma_t}$ implies

$$(p_1 - \epsilon) \cdot \tilde{\gamma}_1(t-1) < \hat{p}_1(\sigma_t)\tilde{\gamma}_1(\sigma_t) < p_{K+1}q_{K+1} + \delta/2,$$

which is equivalent to

$$\tilde{q}_1(t-1) < \frac{p_{K+1}\gamma_{K+1} - (p_1 - \epsilon)(c_1 - c_0) + \delta/2}{(p_1 - \epsilon)(c_2 - c_1)}.$$

Therefore, if $B_t \cap C_{\sigma_t} \cap D_{t-1} \cap D_t \cap E_{\sigma_t}$ happens, we have

$$D_{KL}\left(q_1 - \epsilon, \frac{p_{K+1}\gamma_{K+1} - (p_1 - \epsilon)(c_1 - c_0) + \delta/2}{(p_1 - \epsilon)(c_2 - c_1)}\right) \overset{(a)}{\le} D_{KL}\left(\hat{q}_1(t-1), \frac{p_{K+1}\gamma_{K+1} - (p_1 - \epsilon)(c_1 - c_0) + \delta/2}{(p_1 - \epsilon)(c_2 - c_1)}\right) \le \frac{\log f(t-1)}{T_1^{\text{miss}}(t-1)} \text{ and}$$

$$D_{KL}\left(q_1 + \epsilon, \frac{p_{K+1}\gamma_{K+1} - p_1(c_1 - c_0) + \delta}{p_1(c_2 - c_1)}\right) \ge D_{KL}\left(\hat{q}_1(t), \frac{p_{K+1}\gamma_{K+1} - p_1(c_1 - c_0) + \delta}{p_1(c_2 - c_1)}\right) \ge \frac{\log f(t)}{T_1^{\text{miss}}(t-1) + 1},$$

where the inequality (a) holds because we can choose ϵ small enough such that

$$q_1 - \epsilon > \frac{p_{K+1}\gamma_{K+1} - (p_1 - \epsilon)(c_1 - c_0) + \delta}{(p_1 - \epsilon)(c_2 - c_1)}.$$

As a result, we have

$$\begin{aligned} &\frac{\log f(t-1)}{\log f(t)} \cdot \frac{T_1^{\text{miss}}(t-1)+1}{T_1^{\text{miss}}(t-1)} \\ &\geq \frac{D_{KL}\left(q_1 - \epsilon, \frac{p_{K+1}\gamma_{K+1} - (p_1-\epsilon)(c_1-c_0)+\delta/2}{(p_1-\epsilon)(c_2-c_1)}\right)}{D_{KL}\left(q_1 + \epsilon, \frac{p_{K+1}\gamma_{K+1} - p_1(c_1-c_0)+\delta}{p_1(c_2-c_1)}\right)} \triangleq \eta. \end{aligned} \quad (25)$$

We could select ϵ small enough such that the constant $\eta > 1$.

Lemma 4 implies that the left-hand side of (25) converges to 1 almost surely as $t \to +\infty$. Thus, the probability that (25) happens infinitely often is zero, which indicates $\mathbb{P}[B_t \text{ infinitely often}] = 0$. Combining this with (23) finishes the proof. □

Lemma 6. *Under the KL-LCB policy, we have almost surely,*
1) for $1 \leq i \leq K$, if $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$, then

$$\limsup_{t\to+\infty} \frac{T_i^{miss}(t)}{\log t} \leq 1 \Big/ D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right),$$

if $p_i(c_1 - c_0) \geq p_{K+1}\gamma_{K+1}$, then $\lim_{t\to+\infty} T_i^{miss}(t)/\log t = 0$,
2) for $K+1 \leq i \leq N$, $\lim_{t\to+\infty} T_i^{out}(t)/t = 1$.

Proof of Lemma 6. First, we consider the case for $1 \leq i \leq K$ and $p_i(c_1 - c_0) \geq p_{K+1}\gamma_{K+1}$. Recall that $\tilde{\gamma}_i(t) = \tilde{q}_i(t)c_2 + (1 - \tilde{q}_i(t))c_1 - c_0$. For this case, we have for $\forall t > 0$ and $K+1 \leq j \leq N$, $p_i\tilde{\gamma}_i(t) > p_i(c_1 - c_0) \geq p_{K+1}\gamma_{K+1} \geq p_j\gamma_j$. Based on Lemma 4, we have almost surely that $\limsup_{t\to\infty} \tilde{\gamma}_j(t) \leq \gamma_j$ for $K+1 \leq j \leq N$, implies that for $1 \leq i \leq K$ and $K+1 \leq j \leq N$,

$$\mathbb{P}[\hat{p}_i(t)\tilde{\gamma}_i(t) \leq \hat{p}_j(t)\tilde{\gamma}_j(t) \text{ infinitely often}] = 0.$$

Therefore, d_i will be cached for all but finitely many time slots almost surely, i.e., $\limsup_{t\to+\infty} T_i^{\text{miss}}(t)/\log t = 0$.

Next, we will focus on the case for $1 \leq i \leq K$ and $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$. Define $\underline{D}_{KL}(p, q) = D_{KL}(p, q) \cdot \mathbf{1}(p \geq q)$. According to the definition of $\tilde{q}_i(t)$ in (4), we have

$$\underline{D}_{KL}(\hat{q}_i(t), \tilde{q}_i(t)) = \log f(t)/T_i^{\text{miss}}(t).$$

Thus, Lemma 5 implies that for $\forall\delta > 0$ and $1 \leq i \leq K$,

$$\mathbb{P}\left[\underline{D}_{KL}\left(\hat{q}_i(t), \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) + \delta}{p_i(c_2 - c_1)}\right) \geq \frac{\log f(t)}{T_i^{\text{miss}}(t)} \right. \\ \left. \text{infinitely often}\right] = 0.$$

which, combining with Lemma 4, yields

$$\mathbb{P}\left[\underline{D}_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) + \delta}{p_i(c_2 - c_1)}\right) \geq \frac{\log f(t)}{T_i^{\text{miss}}(t)} \right. \\ \left. \text{infinitely often}\right] = 0.$$

Therefore, we have almost surely, for $1 \leq i \leq K$,

$$\limsup_{t\to\infty} \frac{T_i^{\text{miss}}(t)}{\log t} \leq 1 \Big/ D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right).$$

Next, we will focus on the case for $K+1 \leq i \leq N$. Note that for $K+1 \leq i \leq N$, we have

$$T_i^{\text{in}}(t) \leq \sum_{j=K+1}^{N} T_j^{\text{in}}(t) \leq \sum_{j=1}^{K} T_j^{\text{out}}(t), \quad (26)$$

Combining (26) and the law of large numbers, we have almost surely for $K+1 \leq i \leq N$

$$\begin{aligned} \lim_{t\to+\infty} \frac{T_i^{\text{out}}(t)}{t} &= \lim_{t\to+\infty} \frac{t - T_i^{\text{in}}(t)}{t} \geq 1 - \lim_{t\to+\infty} \frac{\sum_{j=1}^{K} T_j^{\text{out}}(t)}{t} \\ &= 1 - \lim_{t\to+\infty} \frac{\sum_{j=1}^{K} T_j^{\text{miss}}(t)/p_j}{\log t} \cdot \frac{\log t}{t} = 1. \end{aligned}$$

□

Lemma 7. *Under the KL-LCB policy, for $K+1 \leq i \leq N$, we have almost surely $\lim_{t\to\infty} \tilde{q}_i(t) = q_i$.*

Proof of Lemma 7. Based on Lemma 4 and the strong law of large numbers, we have $\lim_{t\to+\infty} \hat{q}_i(t) = q_i$ almost surely. Moreover, Lemma 6 implies that $\lim_{t\to+\infty} \log f(t)/T_i^{\text{miss}}(t) = 0$ almost surely for $K+1 \leq i \leq N$. Thus, we have $\lim_{t\to\infty} \tilde{q}_i(t) = q_i$ almost surely for $K+1 \leq i \leq N$. □

Lemma 8. *Under the KL-LCB policy, for $1 \leq i \leq K, \forall\delta > 0$ and $t = 1, 2, 3, \cdots$, we have*

$$\mathbb{P}\left[\tilde{q}_i(t) \leq \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) - \delta}{p_i(c_2 - c_1)} \textit{ infinitely often}\right] = 0.$$

Proof of Lemma 8. First, if $p_i(c_1 - c_0) \geq p_{K+1}\gamma_{K+1}$, then the proof will be trivial, since $\tilde{q}_i(t)$ must be positive. Next, we consider the case when $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$.

Define the time stamps σ_t and ω_t as follows

$$\begin{aligned} \sigma_t = \min\Bigg\{ & s > t : \text{ For any } 1 \leq i \leq K, \\ & \tilde{q}_i(s) \geq \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) - \delta/2}{p_i(c_2 - c_1)} \Bigg\}, \\ \omega_t = \min\Bigg\{ & s > t : \exists i, 1 \leq i \leq K, \\ & \tilde{q}_i(s) \leq \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) - \delta}{p_i(c_2 - c_1)} \Bigg\}. \end{aligned} \quad (27)$$

Define the event A_t as

$$\begin{aligned} A_t = \Bigg\{ & \text{For any } 1 \leq i \leq K, \\ & \tilde{q}_i(t) \geq \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) - \delta/2}{p_i(c_2 - c_1)} \Bigg\}. \end{aligned}$$

Define the event B_t as

$$B_t = A_t \cap \{\sigma_t > \omega_t\}.$$

According to Lemmas 4 and 6, we know that $\mathbb{P}[A_t \text{ infinitely often}] = 1$. As a result, in order to prove this lemma, it suffices to show

$$\mathbb{P}[B_t \text{ infinitely often}] = 0. \quad (28)$$

Next, we will focus on B_t and prove this result.

Choose $\delta \in (0, p_{K+1}\gamma_{K+1} - p_{K+2}\gamma_{K+2})$. Define the events C_t and D_t as

$$\begin{aligned} C_t =& \{\hat{p}_{K+1}(s)\tilde{\gamma}_{K+1}(s) \in (p_{K+1}\gamma_{K+1} - \delta/4, \\ & p_{K+1}\gamma_{K+1} + \delta/4) \text{ for } \forall t \le s \le t^2\}, \\ D_t =& \{\hat{p}_i(s)\tilde{\gamma}_i(s) \le p_{K+1}\gamma_{K+1} - \delta \\ & \text{for } \forall t \le s \le t^2 \text{ and } \forall K+2 \le i \le N\}. \end{aligned}$$

Lemma 7 and the strong law of large numbers imply that

$$\mathbb{P}[C_t^c \text{ infinitely often}] = \mathbb{P}[D_t^c \text{ infinitely often}] = 0. \quad (29)$$

For a small $\epsilon > 0$, define the events E_t and F_t as

$$\begin{aligned} E_t =& \{\hat{p}_i(s) \in (p_i - \epsilon, p_i + \epsilon) \text{ for } \forall t \le s \le t^2\}, \\ F_t =& \{\hat{q}_i(s) \in (q_i - \epsilon, q_i + \epsilon) \text{ for } \forall t \le s \le t^2\}. \end{aligned}$$

Lemma 4 and the strong law of large numbers imply that

$$\mathbb{P}[E_t^c \text{ infinitely often}] = \mathbb{P}[F_t^c \text{ infinitely often}] = 0. \quad (30)$$

Combining (29), (30) and the fact that $B_t \subseteq \{B_t \cap C_t \cap D_t \cap E_t \cap F_t\} \cup C_t^c \cup D_t^c \cup E_t^c \cup F_t^c$, in order to show (28), it suffices to prove

$$\mathbb{P}[B_t \cap C_t \cap D_t \cap E_t \cap F_t \text{ infinitely often}] = 0. \quad (31)$$

First, assuming that $\{A_t \cap F_t\}$ happens, we will derive a lower bound for ω_t. According to the KL-LCB policy, we have

$$D_{KL}(\hat{q}_i(t), \tilde{q}_i(t)) = \log f(t)/T_i^{\text{miss}}(t)$$

for $1 \le i \le K$ and $f(t)$ defined in (4), which implies

$$\begin{aligned} & D_{KL}\left(q_i + \epsilon, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) - \delta/2}{p_i(c_2 - c_1)}\right) \\ & \ge D_{KL}\left(\hat{q}_i(t), \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) - \delta/2}{p_i(c_2 - c_1)}\right) \\ & \ge \log f(t)/T_i^{\text{miss}}(t). \end{aligned} \quad (32)$$

Based on the definition of ω_t in (27), at the time stamp ω_t, there exists an index j, $1 \le j \le K$ such that

$$\tilde{q}_j(\omega_t) \le \frac{p_{K+1}\gamma_{K+1} - p_j(c_1 - c_0) - \delta}{p_j(c_2 - c_1)}.$$

Therefore, we have

$$\begin{aligned} & D_{KL}\left(q_j - \epsilon, \frac{p_{K+1}\gamma_{K+1} - p_j(c_1 - c_0) - \delta}{p_j(c_2 - c_1)}\right) \\ & \le D_{KL}\left(\hat{q}_j(\omega_t), \frac{p_{K+1}\gamma_{K+1} - p_j(c_1 - c_0) - \delta}{p_j(c_2 - c_1)}\right) \\ & \le \frac{\log f(\omega_t)}{T_j^{\text{miss}}(\omega_t)} \le \frac{\log f(\omega_t)}{T_j^{\text{miss}}(t)}. \end{aligned} \quad (33)$$

Define

$$\eta_j = \frac{D_{KL}\left(q_j - \epsilon, \frac{p_{K+1}\gamma_{K+1} - p_j(c_1 - c_0) - \delta}{p_j(c_2 - c_1)}\right)}{D_{KL}\left(q_j + \epsilon, \frac{p_{K+1}\gamma_{K+1} - p_j(c_1 - c_0) - \delta/2}{p_j(c_2 - c_1)}\right)}.$$

Combining (32) and (33) yields, for $1 \le j \le K$,

$$\frac{\log f(\omega_t)}{\log f(t)} = \frac{\log(1 + \omega_t(\log \omega_t)^2)}{\log(1 + t(\log t)^2)} \ge \eta_j.$$

Define $\eta = \min_{1 \le j \le K} \eta_j$. We can select ϵ small enough, such that $\eta > 1$. Let $t = \tau_1$ be the unique solution to $(\log t)^\eta = \eta \log t$. For $t > \tau_1$, ω_t could be lower bounded by

$$\omega_t > t^\eta.$$

Next, we will show that if there are sufficiently many requests for d_i, $1 \le i \le K$, arriving during the time interval $[t, t^\eta]$, then the event $\{B_t \cap C_t \cap D_t \cap E_t \cap F_t\}$ will not happen. In particular, select $\theta_{i,t}$ as

$$\theta_{i,t} = \left\lceil \frac{\log f(t^\eta)}{D_{KL}\left(q_i - \epsilon, \frac{p_{K+1}\gamma_{K+1} - p_j(c_1 - c_0) - \delta/2}{p_i(c_2 - c_1)}\right)} \right\rceil.$$

Assume that $\{C_t \cap D_t \cap E_t \cap F_t\}$ happens. We can select δ small enough such that $\eta \in (1, 2)$. So the time interval $[t, t^\eta]$ could be covered by the interval $[t, t^2]$ considered in the events C_t, D_t, E_t, F_t. If there are $\theta_{i,t}$ misses happen for d_i, $1 \le i \le K$, in the time interval $[t, t^\eta]$, we have

$$\begin{aligned} & D_{KL}\left(q_i - \epsilon, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) - \delta/2}{p_i(c_2 - c_1)}\right) \\ & \quad \ge \frac{\log f(t^\eta)}{\theta_{i,t}} \ge \frac{\log f(t^\eta)}{T_i^{\text{miss}}(t^\eta)}. \\ & \Rightarrow D_{KL}\left(\hat{q}_i(t^\eta), \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) - \delta/2}{p_i(c_2 - c_1)}\right) \\ & \quad \ge \frac{\log f(t^\eta)}{T_i^{\text{miss}}(t^\eta)}. \\ & \Rightarrow \tilde{q}_i(t^\eta) \ge \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0) - \delta/2}{p_i(c_2 - c_1)}. \end{aligned}$$

Therefore, if there are $\theta_{i,t}$ misses for each d_i, $1 \le i \le K$, in the time interval $[t, t^\eta]$, then we will have $\sigma_t \le t^\eta < \omega_t$, i.e., the event $\{B_t \cap C_t \cap D_t \cap E_t \cap F_t\}$ will not happen.

Leveraging this result, we could divide the time interval $[t, t^\eta]$ into $\sum_{i=1}^K \theta_{i,t} + 1$ small chunks with equal length. It can be verified that $\{B_t \cap C_t \cap D_t \cap E_t \cap F_t\}$ will not happen, if d_{K+1} is requested in the first chunk and data items d_i, $1 \le i \le K$, are requested at least once in each of the remaining chunks. This claim can be verified based on the following observations. When $\{C_t \cap D_t \cap E_t \cap F_t\}$ happens, we have

1) d_i, $K+2 \le i \le N$ will be cache misses and never loaded into the cache in the time interval $[t, t^\eta]$ because of D_t.
2) If d_{K+1} is not loaded into the cache when it is requested in the first chunk, or if it is loaded into the cache but then evicted before t^η, let t' denote the timestamp when this happens and we must have $\sigma_t \le t' \le t^\eta < \omega_t$, i.e., the event $\{B_t \cap C_t \cap D_t \cap E_t \cap F_t\}$ does not happen.
3) If the condition in 2) does not happen, then in each chunk except the first one, a cache miss must occur for some d_i, $1 \le i \le K$.
4) If for some d_i, $1 \le i \le K$, there are more than $\theta_{i,t}$ misses happen in the interval $[t, t^\eta]$, then let t' denote the timestamp when the $(\theta_{i,t} + 1)$'th miss happens. We have $\sigma_t \le t' \le t^\eta < \omega_t$, i.e., the event $\{B_t \cap C_t \cap D_t \cap E_t \cap F_t\}$ will not happen.
5) If the condition in 4) does not happen, then based on 3), each d_i, $1 \le i \le K$ will have exactly $\theta_{i,t}$ misses in $[t, t^\eta]$, and therefore, the event $\{B_t \cap C_t \cap D_t \cap E_t \cap F_t\}$ will not happen.

In other words, if the event $\{B_t \cap C_t \cap D_t \cap E_t \cap F_t\}$ happens, then either d_{K+1} is not requested in the first chunk, or there exist a data item d_i, $1 \le i \le K$, that is not requested in at least one of the remaining chunks. With this conclusion, we can derive the following bound

$$\begin{aligned}
&\mathbb{P}[B_t \cap C_t \cap D_t \cap E_t \cap F_t] \\
&\le \mathbb{P}[d_{K+1} \text{ is not requested in the first chunk}] \\
&\quad + \mathbb{P}\Bigg[\text{Exist some } 1 \le i \le K, 2 \le m \le \sum_{j=1}^{K} \theta_{j,t} + 1, \\
&\qquad\qquad \text{such that } d_i \text{ is not requested in the } m^{th} \text{ chunk}\Bigg] \\
&\le (1 - p_{K+1})^{l(t)} + \sum_{i=1}^{K} (1 - p_i)^{l(t)} \cdot \sum_{i=1}^{K} \theta_{i,t}.
\end{aligned}$$

where $l(t) = (t^{\eta} - t)/(\sum_{i=1}^{K} \theta_{j,t} + 1)$ is the length of each time chunk. Since $\eta > 1$ and $\theta_{i,t} = O(\log t)$, we have

$$\sum_{t=1}^{+\infty} \mathbb{P}[B_t \cap C_t \cap D_t \cap E_t \cap F_t] < +\infty,$$

which, based on the Borel-Cantelli Lemma, implies (31) and completes the proof. □

Lemma 9. *Under KL-LCB based policy, for* $1 \le i \le K$, *if* $p_i(c_1 - c_0) < p_{K+1}\gamma_{K+1}$, *then we have almost surely*

$$\liminf_{t \to +\infty} \frac{T_i^{miss}(t)}{\log t} \ge 1 \Big/ D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right).$$

The proof of Lemma 9 is similar to the proof of Lemma 6, and is omitted due to the page limit.

Now we are ready to prove Lemma 3. First, we have

$$\lim_{t \to +\infty} \tilde{q}_i(t) = \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}, \tag{34}$$

for $1 \le i \le K$ almost surely due to Lemmas 5 and 8. Note that the KL-LCB based policy attempts to cache the data items with the largest $\hat{p}_i(t)\tilde{\gamma}_i(t)$. Thus, combining Lemmas 6, 7, 9, Equation (34) and the strong law of large numbers, we have

$$\begin{aligned}
\lim_{t \to +\infty} \frac{T_{K+1}^{\text{in}}(t)}{\log t} &= \lim_{t \to +\infty} \frac{\sum_{i=1}^{K} T_i^{\text{out}}(t)}{\log t} \\
&= \lim_{t \to +\infty} \frac{\sum_{i=1}^{K} T_i^{\text{miss}}(t)/p_i}{\log t} \\
&= \sum_{i=1}^{K} \frac{1}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right) p_i},
\end{aligned}$$

almost surely. Convergence almost surely implies convergence in probability. Therefore, we have, for $\forall \epsilon \in (0, 1)$,

$$\begin{aligned}
&\lim_{t \to \infty} \mathbb{P}\left[\frac{T_{K+1}^{\text{in}}(t)}{\log t} \ge \sum_{i=1}^{K} \frac{1 - \epsilon}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right) p_i}\right] \\
&= 1.
\end{aligned} \tag{35}$$

Therefore, we have for $\forall \epsilon \in (0, 1)$,

$$\begin{aligned}
&\frac{\mathbb{E}\left[T_{K+1}^{\text{in}}(t)\right]}{\log t} \\
&\ge \mathbb{P}\left[\frac{T_{K+1}^{\text{in}}(t)}{\log t} \ge \sum_{i=1}^{K} \frac{1 - \epsilon}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right) p_i}\right] \\
&\quad \cdot \mathbb{E}\left[\frac{T_{K+1}^{\text{in}}(t)}{\log t} \middle| \frac{T_{K+1}^{\text{in}}(t)}{\log t}\right. \\
&\qquad\qquad \left. \ge \sum_{i=1}^{K} \frac{1 - \epsilon}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right) p_i}\right] \\
&\ge \mathbb{P}\left[\frac{T_{K+1}^{\text{in}}(t)}{\log t} \ge \sum_{i=1}^{K} \frac{1 - \epsilon}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right) p_i}\right] \\
&\quad \cdot \sum_{i=1}^{K} \frac{1 - \epsilon}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right) p_i}.
\end{aligned} \tag{36}$$

Combining (35) and (36) yields

$$\liminf_{t \to +\infty} \frac{\mathbb{E}\left[T_{K+1}^{\text{in}}(t)\right]}{\log t} \ge \sum_{i=1}^{K} \frac{1}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right) p_i}.$$

Furthermore, it is easy to observe that $T_{K+1}^{\text{in}}(t) \le \sum_{i=1}^{K} T_i^{\text{out}}(t)$, because at any time there are at most K data items are cached. Thus, combining Lemma 2 and the strong law of large numbers, we have

$$\begin{aligned}
\limsup_{t \to +\infty} \frac{\mathbb{E}\left[T_{K+1}^{\text{in}}(t)\right]}{\log t} &\le \limsup_{t \to +\infty} \sum_{i=1}^{K} \frac{\mathbb{E}\left[T_i^{\text{out}}(t)\right]}{\log t} \\
&\le \sum_{i=1}^{K} \frac{1}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right) p_i}.
\end{aligned}$$

Combining the upper and the lower bounds yields

$$\lim_{t \to +\infty} \frac{\mathbb{E}\left[T_{K+1}^{\text{in}}(t)\right]}{\log t} = \sum_{i=1}^{K} \frac{1}{D_{KL}\left(q_i, \frac{p_{K+1}\gamma_{K+1} - p_i(c_1 - c_0)}{p_i(c_2 - c_1)}\right) p_i},$$

which, together with Lemma 2 and the fact that $\sum_{i=1}^{K} T_i^{\text{out}}(t) \ge T_{K+1}^{\text{in}}(t)$, finishes the proof of Lemma 3.

References

[1] Q. Huang, K. Birman, R. Van Renesse, W. Lloyd, S. Kumar, and H. C. Li, "An analysis of Facebook photo caching," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, 2013, pp. 167–181.

[2] L. Cherkasova, *Improving WWW proxies performance with greedy-dual-size-frequency caching policy*. Hewlett-Packard Laboratories, 1998.

[3] P. Cao and S. Irani, "Cost-aware WWW proxy caching algorithms." in *Usenix symposium on internet technologies and systems*, vol. 12, no. 97, 1997, pp. 193–206.

[4] C. Li and A. L. Cox, "GD-Wheel: a cost-aware replacement policy for key-value stores," in *Proceedings of the Tenth European Conference on Computer Systems*, 2015, pp. 1–15.

[5] N. Young, *Competitive paging and dual-guided on-line weighted caching and matching algorithms*. Princeton University, 1991.

[6] A. Blankstein, S. Sen, and M. J. Freedman, "Hyperbolic caching: Flexible caching for web applications," in *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, 2017, pp. 499–511.

[7] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[8] I. Szita and A. Lőrincz, "The many faces of optimism: a unifying approach," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1048–1055.

[9] P. Cao and S. Irani, "GreedyDual-Size: A cost-aware WWW proxy caching algorithm," in *2nd Web Caching Workshop, Boulder, Colorado*, 1997.

[10] B. Hou and F. Chen, "GDS-LC: A latency-and cost-aware client caching scheme for cloud storage," *ACM Transactions on Storage (TOS)*, vol. 13, no. 4, pp. 1–33, 2017.

[11] J. Shim, P. Scheuermann, and R. Vingralek, "Proxy cache algorithms: Design, implementation, and performance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 4, pp. 549–562, 1999.

[12] S. Liang, K. Chen, S. Jiang, and X. Zhang, "Cost-aware caching algorithms for distributed storage servers," in *International Symposium on Distributed Computing*. Springer, 2007, pp. 373–387.

[13] J. Song, M. Sheng, T. Q. Quek, C. Xu, and X. Wang, "Learning-based content caching and sharing for wireless networks," *IEEE Transactions on Communications*, vol. 65, no. 10, pp. 4309–4324, 2017.

[14] A. Sengupta, S. Amuru, R. Tandon, R. M. Buehrer, and T. C. Clancy, "Learning distributed caching strategies in small cell networks," in *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*. IEEE, 2014, pp. 917–921.

[15] X. Xu, M. Tao, and C. Shen, "Collaborative multi-agent multi-armed bandit learning for small-cell caching," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2570–2585, 2020.

[16] A. Bura, D. Rengarajan, D. Kalathil, S. Shakkottai, and J.-F. Chamberland, "Learning to cache and caching to learn: Regret analysis of caching algorithms," *IEEE/ACM Transactions on Networking*, 2021.

[17] P. Blasco and D. Gündüz, "Multi-armed bandit optimization of cache content in wireless infostation networks," in *2014 IEEE International Symposium on Information Theory*. IEEE, 2014, pp. 51–55.

[18] ——, "Learning-based optimization of cache content in a small cell base station," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 1897–1903.

[19] A. Sadeghi, F. Sheikholeslami, A. G. Marques, and G. B. Giannakis, "Reinforcement learning for adaptive caching with dynamic storage pricing," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2267–2281, 2019.

[20] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting caching and multicast for 5G wireless networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2995–3007, 2016.

[21] Y. Cui and D. Jiang, "Analysis and optimization of caching and multicasting in large-scale cache-enabled heterogeneous wireless networks," *IEEE transactions on Wireless Communications*, vol. 16, no. 1, pp. 250–264, 2016.

[22] B. Zhou, Y. Cui, and M. Tao, "Optimal dynamic multicast scheduling for cache-enabled content-centric wireless networks," *IEEE Transactions on Communications*, vol. 65, no. 7, pp. 2956–2970, 2017.

[23] M. M. Amiri and D. Gündüz, "Caching and coded delivery over Gaussian broadcast channels for energy efficiency," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1706–1720, 2018.

[24] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Delay gain analysis of wireless multicasting for content distribution," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 529–542, 2020.

[25] G. Quan, A. Eryilmaz, and N. B. Shroff, "Optimal edge caching for individualized demand dynamics," *IEEE/ACM Transactions on Networking*, pp. 1–16, 2024.

[26] G. Quan, J. Tan, and A. Eryilmaz, "Counterintuitive characteristics of optimal distributed LRU caching over unreliable channels," *IEEE/ACM Transactions on Networking*, vol. 28, no. 6, pp. 2461–2474, 2020.

[27] S. Vanichpun and A. M. Makowski, "The output of a cache under the independent reference model: where did the locality of reference go?" in *Proceedings of the joint international conference on Measurement and modeling of computer systems*, 2004, pp. 295–306.

[28] R. Fagin and T. G. Price, "Efficient calculation of expected miss ratios in the independent reference model," *SIAM Journal on Computing*, vol. 7, no. 3, pp. 288–297, 1978.

[29] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[30] T. L. Lai, "Adaptive treatment allocation and the multi-armed bandit problem," *The Annals of Statistics*, pp. 1091–1114, 1987.

[31] O. Cappé, A. Garivier, O.-A. Maillard, R. Munos, and G. Stoltz, "Kullback-Leibler upper confidence bounds for optimal sequential allocation," *The Annals of Statistics*, pp. 1516–1541, 2013.

[32] D. Yun, A. Proutiere, S. Ahn, J. Shin, and Y. Yi, "Multi-armed bandit with additional observations," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 1, pp. 1–22, 2018.

[33] J. Zuo, X. Zhang, and C. Joe-Wong, "Observe before play: Multi-armed bandit with pre-observations," *ACM SIGMETRICS Performance Evaluation Review*, vol. 46, no. 2, pp. 89–90, 2019.

[34] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari, "Tight regret bounds for stochastic combinatorial semi-bandits," in *Artificial Intelligence and Statistics*. PMLR, 2015, pp. 535–543.

[35] R. Degenne and V. Perchet, "Combinatorial semi-bandit with known covariance," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[36] Y. Yang and J. Zhu, "Write skew and Zipf distribution: Evidence and implications," *ACM transactions on Storage (TOS)*, vol. 12, no. 4, pp. 1–19, 2016.

[37] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, vol. 1. IEEE, 1999, pp. 126–134.

Guocong Quan received the Ph.D. degree in electrical and computer engineering from The Ohio State University in 2021. Then he joined Meta as a research scientist. His research interest focuses on resolving challenges in distributed networking and computing systems. He received the 2019 IEEE INFOCOM Best Paper Award.

Atilla Eryilmaz (Senior Member, IEEE) 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 worked as a Post-Doctoral Associate at 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 and the Graduate Studies Chair of the Electrical and Computer Engineering Department. His research interests include optimal control of stochastic networks, machine learning, optimization, and information theory. He received the NSF-CAREER Award in 2010 and the two Lumley Research Awards for Research Excellence in 2010 and 2015. He is a coauthor of the 2012 IEEE WiOpt Conference Best Student Paper, subsequently received the 2016 IEEE INFOCOM Best Paper Award, the 2017 IEEE WiOpt Best Paper Award, the 2018 IEEE WiOpt Best Paper Award, and the 2019 IEEE INFOCOM Best Paper Award. He has served as a TPC Co-Chair for IEEE WiOpt in 2014, ACM MobiHoc in 2017, and IEEE INFOCOM in 2022; and an Associate Editor for IEEE/ACM Transactions on Networking from 2015 to 2019 and IEEE Transactions on Network Science and Engineering from 2017 to 2022. He has been an Associate Editor of the IEEE Transactions on Information Theory, since 2022.

Ness B. Shroff (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA, in 1994. He joined Purdue University, West Lafayette, IN, USA, immediately thereafter as an Assistant Professor with the School of Electrical and Computer Engineering. At Purdue, he became a Full Professor of ECE and the Director of a University-Wide Center on Wireless Systems and Applications in 2004. In 2007, he joined The Ohio State University, Columbus, OH, USA, where he holds the Ohio Eminent Scholar Endowed Chair in networking and communications, with the Departments of ECE and CSE. He is currently the Institute Director of the NSF AI Institute for Future Edge Networks and Distributed Intelligence. He was the recipient of numerous best paper awards for his research and is listed in Thomson Reuters' on The World's Most Influential Scientific Minds, and has been noted as a Highly Cited Researcher by Thomson Reuters in 2014 and 2015. He also was the recipient of the IEEE INFOCOM Achievement Award for seminal contributions to scheduling and resource allocation in wireless networks.