Optimal Edge Caching For Individualized Demand Dynamics

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

***Abstract*—The ever-growing end user data demands, and the reductions in memory costs are fueling edge-caching deployments. Caching at the edge is substantially different from that at the core and needs to consider the nature of individualized data demands. For example, an individual user may not be interested in requesting the same data item again, if it has recently requested it. Such individualized dynamics are not apparent in the aggregated data requests at the core and have not been considered in popularity-driven caching designs for the core. Hence, these traditional caching policies could induce significant inefficiencies when applied at the edges. To address this issue, we develop new edge caching policies optimized for the individualized demands that also leverage overhearing opportunities at the wireless edge. With the objective of maximizing the hit ratio, the proposed policies will actively evict the data items that are not likely to be requested in the near future, and strategically bring them back into the cache via overhearing when they become popular again. Both theoretical analysis and numerical simulations demonstrate that the proposed edge caching policies could outperform the popularity-driven policies that are optimal at the core.**

***Index Terms*—edge caching, broadcasting, overhearing**

I. Introduction

Data demands are growing exponentially, driven by the rapid proliferation of edge devices such as the Internet of Things (IoT), and increasingly capable hand-held devices. Meanwhile, memory is becoming cheaper, larger, and faster. These two forces are creating an ideal environment for the large-scale deployment of edge caching to support fast data retrieval [1]–[4]. While, extensive studies [5]–[8] have been conducted to optimize caching strategies for relatively stationary data demands at the network core, caching at the edges due to its individualized demand dynamics, is quite different from the core, and therefore should be studied in their own right. In this paper, we will propose new caching policies optimized for the individualized data demands at the wireless edges.

A. Challenge: Individualized Demands at Network Edges

At the network core, data demands are aggregated from a large number of end-users, as shown in Fig. 1. Thus, the demand dynamics of each individual user could become negligible, which leads to relatively stationary data popularities for the population. Various popularity-driven policies have been proposed for optimizing caching at the core [6], [9]. Inspired by the observation that data items recently requested by one user are very likely to be requested again by others, the least recently used (LRU) policy estimates the popularity by the data recency and caches the most recently requested data items. The LRU policy and its variants have been widely implemented at the core, and validated to achieve good performance [10]–[12].

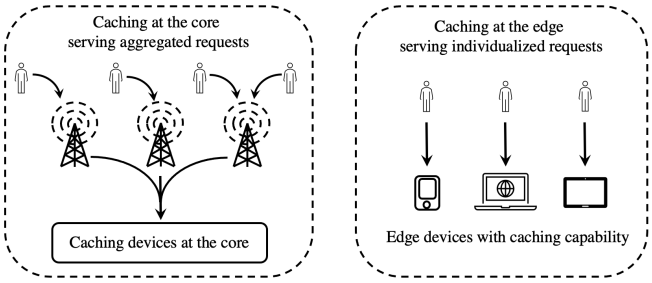


Fig. 1: Caching at the core v.s. caching at the edge.

In contrast, edge caching serves a small group of users, or even a single user, where the data demands are more individualized. Those have fundamentally different dynamics than the population demand models. In particular, *after requesting a data item, the user may not be likely to request the same data item again in the near future*. One supportive reason is that users may lose interest in seeing similar content repeatedly. A common methodology applied by recommendation systems is to avoid presenting similar content consecutively [13]–[15]. Another evidence is that users may need some time to process the recently requested data. A trace analysis on Yelp has demonstrated considerable time gaps between users' actions [16]. Hence, data popularities at the edges could change dramatically even after every request. The popularity-driven policies designed for the core cannot make adaptive decisions for such individualized demands, and may achieve poor performance at the edges.

In Fig. 2, we illustrate an example showing that the LRU policy could make irrational decisions for edge caching, since the recently requested data items typically have small popularities at the edges, which is opposite to the experience at the core. Assume that an individual user will not request the same data item in the near future after each request[1].

The authors are with The Ohio State University, Columbus, OH 43210, USA (emails: {quan.72, eryilmaz.2, shroff.11}@osu.edu). This work was supported by NSF grants NSF AI Institute (AI-EDGE) 2112471, CNS-NeTS-2106679, CNS-NeTS-2007231, CNS-2312836, CNS-2223452, CNS-2225561, CNS-2106933, CNS-2106932, an ONR Grant N00014-19-1-2621, a grant from the Army Research Office W911NF-21-1-0244, and was sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF-23-2-0225. 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.

[1]The data requests are generated with $N = 1000, b = 50, s_i = 5000, \beta_i = c \cdot i^{-1.4}, c = 1/\sum_{i=1}^{N} i^{-1.4} = 0.3392, 1 \leq i \leq N$, where the detailed parameter definitions are introduced in Section II-A

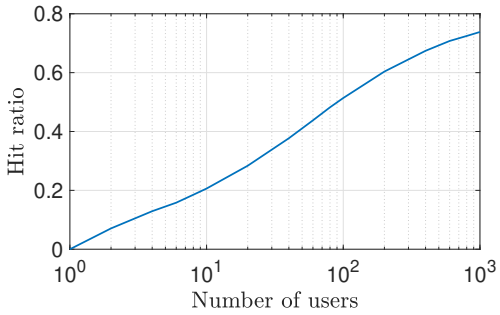


Fig. 2: Degenerate performance of LRU when serving a small group of users.

We simulate the hit ratio achieved by a single LRU cache serving aggregated data requests from a group of users. The LRU policy achieves good performance when the number of users is large (i.e., the network core scenario). However, the performance degenerates significantly as the number of users decreases (i.e., the network edge scenario). Interestingly, when the cache serves one user, the hit ratio will decrease to zero, which indicates that the LRU policy almost always makes the wrong decisions.

B. Solution: Active Eviction and Strategical Overhearing

To address this issue, we develop new adaptive edge caching policies customized for the individualized demands. In an ideal case, the policy should frequently update the cache content and only store data items that are most likely to be requested in the near future. We will leverage the overhearing opportunities at the wireless edges to mimic this ideal design.

Specifically, an edge cache can overhear the broadcasted data items over the wireless channels, even when it is not the intended receiver. To achieve high caching efficiency, we may actively evict the recently requested data items that will not be needed in the near future, and strategically bring them back into the cache later through overhearing when their popularities rise up sufficiently again. With the objective to maximize the overall hit ratio, we optimize the eviction and overhearing decisions for two different settings depending on how the overhearing opportunities are generated. Under the time-driven overhearing setting (cf. Section III), the overhearing opportunities are described by Poisson processes that are independent of the data requests and out of the designer's control. Under the event-driven overhearing setting (cf. Section IV), the overhearing opportunities are generated when an item is requested and is unavailable at a user, which triggers its broadcast over the wireless channel, hereby generating an overhearing opportunity for all other users. Our contributions are summarized as follows.

- With the objective to maximize the overall hit ratio, we propose an optimal caching and overhearing policy for the time-driven overhearing setting. Specifically, we first prove that the hit ratio maximization problem is nonconvex. By exploiting an informative structure of the optimal solutions, we then convert the nonconvex problem to a convex one and propose efficient algorithms to solve it (see Section III).
- We propose an asymptotically optimal caching and overhearing policy for the event-driven overhearing setting. Although the overhearing process is not fully tractable under this setting, we are inspired by the structure of the optimal policy under time-driven overhearing and propose a policy for event-driven overhearing, which is asymptotically optimal when the number of edge caches in the system is sufficiently large (see Section IV).
- We extend our main results for both time-driven and event-driven overhearing scenarios to a more general data demand setting, where different users can have heterogeneous demand patterns (see Section V).
- We conduct extensive simulations to validate that the proposed policies can achieve better performance than a few benchmarks (see Section VI).

C. Related Works

Conventional caching analysis for stationary data demands typically assumes an independent reference model (IRM), where the data requests are assumed to be generated from a stationary popularity distribution independently. Popularity driven caching policies are proposed for such scenarios in different systems [6], [17]. Historical request information including data recency and frequency are commonly leveraged to estimate the popularity, and inspire the design of LRU, LFU, LIRS and other variants [9], [18]–[21]. Among the various caching policies, the time-to-live (TTL) based policies have garnered significant attention, since they are not only easy to implement in real practice, but also provide tractability and flexibility to optimize different system goals [22]–[25]. However, how to design a good TTL-based policy for edge caching with individualized demand dynamics still lacks a systematic study.

To characterize the non-stationary data demands whose popularities may evolve over time, a shot noise model (SNM) is proposed in [26], where the request process of a data item is described by a time-inhomogeneous Poisson process. Compared to IRM, SNM could better characterize the temporal locality and is validated by real data traces collected from more than 10000 IPs. However, under the general SNM, the theoretical analysis of some caching strategies may become intractable. To address this issue, an ON-OFF traffic model is proposed in [27], which captures time-variant data popularities and supports efficient analysis for a number caching strategies. An age-based threshold (ABT) caching policy is proposed for small user populations under SNM [28].

Numerous studies have explored methods for tracking dynamic data demands and optimizing caching decisions to achieve better efficiency [29]–[33]. However, they are different from this paper in the following aspects. 1) Existing works typically consider the aggregated demands from a group of users and attempt to track the dynamic demands over a relatively long time period by collecting historical requests from different users. The individualized demands that could change dramatically in a short time period (e.g., after each request for the data item) have not been well addressed by existing works. 2) These works did not explore the joint optimization

of overhearing and caching decisions when demands are dynamically changing. In this paper, we fill the gap by proposing a new model to describe the individualized demand dynamics and designing new edge caching policies that achieve provably better performance by leveraging overhearing opportunities at the wireless edges.

Another prevalent category of dynamics in caching problems is the content dynamics, where each data item could be occasionally refreshed, rendering older versions obsolete. Different caching strategies have been explored to optimize caching efficiency and content freshness [34]–[36]. We note that such content dynamics are different from the demand dynamics investigated in this paper, because the data popularity changes under content dynamics are triggered by the refresh of content sources, while popularities under demand dynamics are changing with users' actions (e.g., recent requests). Furthermore, edge caching with overhearing opportunities have been shown great potential to improve energy efficiency, transmission delays and data freshness [37]–[41], but these designs didn't consider the individualized demand dynamics.

II. Model Description

In this section, we will first formally model the edge demand dynamics by ON-OFF processes. Then we will introduce TTL based caching and overhearing policies and formulate a hit ratio maximization problem.

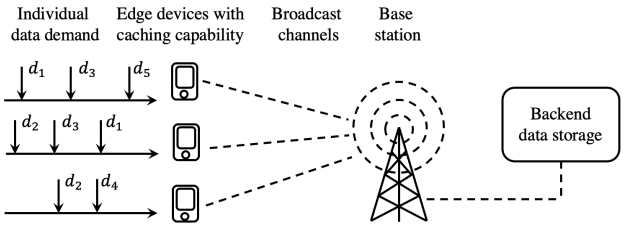


Fig. 3: Edge caching with individual data demand.

A. Individual Demand Dynamics

Consider M edge caches connected to a base station through wireless channels, as shown in Fig. 3. Each edge cache serves data requests from a single user. We use m, $1 \leq m \leq M$, to index an edge cache or the user served by the edge cache interchangeably. Let $\{d_i,\ 1 \leq i \leq N\}$ denote a set of N distinct data items. Assume that the data items are of unit size and each edge cache has a size of b, $0 < b \leq N$. Each edge cache serves an individual user independently. If the requested data is stored in the cache, then the request could be served immediately with a low latency, which is called a cache hit. Otherwise, the requested data has to be obtained from the backend data storage and sent back to serve the user's demand, which is called a cache miss.

To characterize the demand dynamics of individual users, we model the requests for the data item d_i, $1 \leq i \leq N$, generated by each user as a renewable ON-OFF process. Specifically, after the user requests d_i, he/she will not request it again within s_i units of time, which is the OFF period. The OFF period can effectively capture the transfer of user interests as well as the time gaps between users' actions as demonstrated by a trace analysis on Yelp [16]. After the OFF period, the next request for d_i will be generated based on a Poisson process with rate β_i, which is the ON period. Without loss of generality, we assume that the data items are indexed such that β_i's are decreasing with respect to i. When a new request is generated in the ON period, a subsequent OFF period starts immediately and the ON-OFF process is renewed.

For example, Fig. 4 illustrates a sequence of requests for the data item d_1 with $s_1 = 2$ and $\beta_1 = 1$. The first request for d_1 is initiated at epoch 0. After the first request, there is an OFF period of a fixed duration 2, during which, the user is not interested in requesting d_1. Starting from epoch 2, the first OFF period ends, ushering in an ON period, where the user would request d_1 again. Within the ON period, the next request for d_1 will be generated based on a Poisson process with a rate $\beta_1 = 1$. In this example, the next request occurs at epoch 4, at which time, the second OFF period starts and the whole process is renewed.

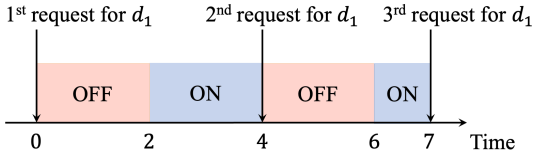


Fig. 4: Individualized demands characterized by renewable ON-OFF processes.

The proposed renewable ON-OFF process describes how a user's demand for a data item will evolve based on the user's recent requests for it. The proposed model could characterize different demand patterns depending on the value of s_i

- When $0 < s_i < +\infty$, the data item won't be requested again in the near future, if the user has recently made a request for it. However, over time, the user might regain interest in it. For example, music within a playlist typically follow this demand pattern.
- When $s_i = +\infty$, d_i will never be requested again after the first request for it. For example, the weather information for today is rarely in demand beyond the current day.
- When $s_i = 0$, the requests for data item d_i will be generated following a Poisson arrival process with a constant rate β_i, which indicates that users are consistently interested in such data items. This setting corresponds to the conventional accumulated demand patterns at the network core.

In this paper, we assume that the parameters s_i and β_i are fixed and known, and focus on how we should update the edge cache content for a set of candidate data items with different demand patterns (i.e., different β_i, s_i values). In real practice, s_i and β_i could be unknown and different approaches could be applied to estimate them. As an illustration, we can employ clustering algorithms to group users with similar interests, enabling us to leverage the observed historical requests from similar users to estimate parameters for others [29].

In the main paper, we consider the homogeneous demand dynamics, where different users have the same request pattern (i.e., s_i, β_i only depend on the data item d_i and are identical

for different users). This setting is particularly relevant to scenarios where individual users have common data interests (e.g., on-trend music and TV series). The homogeneous setting helps us focus on the impact of individualized demands at the network edge as opposed to the aggregated demands at the network core. Later, in Section V, we will show that most of the theorems and insights obtained for the homogeneous setting are also valid when users have heterogeneous demands.

B. Overhearing Opportunities

To improve the edge caching efficiency under such dynamic data demands, we will leverage overhearing opportunities over the wireless channels. Specifically, the base station can broadcast data items from time to time. When the cache overhears a data item, it may decide to store it or not based on the adopted caching and overhearing policies. Since the users are assumed to have common data interests, the overheard data item could potentially satisfy the demand of multiple users, and therefore, improves the caching efficiency. Note that the privacy concerns or the data encryption are not considered in this paper. Developing efficient edge caching policies with privacy protection is a crucial avenue for future research. However, this topic falls outside the scope of this paper.

In this paper, we will investigate optimal caching and overhearing policy under the following two different overhearing scenarios depending on how the overhearing opportunities are generated.

Scenario 1 (Time-driven overhearing): In this scenario, we assume that the base station will broadcast each data item based on independent Poisson processes with given fixed rates. The caches could passively overhear and need to decide whether to store the overheard data items. Note that the overhearing processes in this scenario are fixed and independent of caching decisions. The time-driven overhearing opportunities are given by the environment and our goal is to find good policies to leverage these opportunities. This simple setting could provide us informative insights to optimize caching decisions with overhearing opportunities, which inspires the policy design for more realistic settings in Scenario 2.

Scenario 2 (Event-driven overhearing): In this scenario, we consider a more realistic setting. When a cache miss happens for some data item, e.g., d_1, the base station will fetch d_1 from the backend storage and send it back to the cache over the broadcast channel. Meanwhile, the other caches could overhear d_1 and decide whether to cache it or not. Unlike time-driven overhearing, the event-driven overhearing opportunities are not fixed or given by the environment. They are shaped by the miss behaviors, which, in turn, are determined by the caching policies. Thus, the policy design for this scenario is more challenging.

C. TTL-Based Caching and Overhearing Policies

To achieve as many hits as possible, the key question to answer is how to update the cache content. In this paper, we consider the design where each data item can be updated separately based on its own rule. Formally, we use a vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_N)$ to denote the policy for managing the N data items, where each element π_i is the update rule for a data item d_i, $1 \leq i \leq N$. To avoid possible confusion, we call $\boldsymbol{\pi}$ a policy and each π_i an item policy.

First, consider the item policies that belong to the following TTL based caching and overhearing item policy set. Define

$$\Pi^{\text{co}} = \{\pi^{\text{co}}(\tau, \omega) : \omega \geq \tau \geq 0\}, \tag{1}$$

where an item policy $\pi^{\text{co}}(\tau, \omega)$ is determined by two parameters, i.e., the caching TTL τ and the deaf TTL ω. The superscript "co" stands for caching and overhearing.

In particular, assume that the data item d_i is served by the item policy $\pi^{\text{co}}(\tau_i, \omega_i)$. Then every time the data item d_i is requested, it will be loaded into the cache regardless of a hit or a miss. Meanwhile, a caching timer with duration τ_i and a deaf timer with duration ω_i ($\omega_i \geq \tau_i$) will be initiated. In Fig. 5, we illustrate an item policy $\pi^{\text{co}}(\tau_1, \omega_1)$ for d_1 with $\tau_1 = 3$ and $\omega_1 = 7$.

1) Until the caching timer expires, d_i will be cached but promptly evicted once the timer runs out. In Fig. 5, d_1 will be cached during the period [0, 3] and evicted at epoch 3.
2) Before the deaf timer expires, the cache refrains from loading d_i into cache via overhearing. In Fig. 5, although d_1 is broadcasted at epoch 5, it will not be loaded into cache based on the chosen policy.
3) After the deaf timer expires, the cache will opportunistically store d_i via overhearing when it is broadcasted. Once a request for d_i is generated and fulfilled, both two timers will be reset, and the procedure will be renewed to serve the next request. In Fig. 5, d_1 will be overheard and loaded into the cache at epoch 8, if the second request for d_1 is not generated before epoch 8.

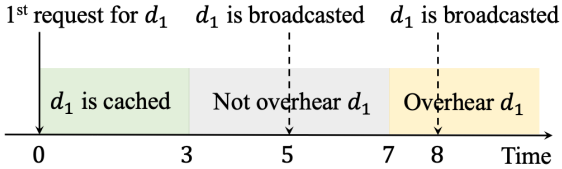


Fig. 5: TTL-based caching and overhearing policy.

It is easy to observe that whether the next request for d_i is a hit or a miss depends on when it arrives. We still use the example in Fig. 5 to illustrate this process.

1) Request before eviction: If the second request for d_1 arrives before the caching timer expires (i.e., epoch 3), then it is a cache hit since d_1 has not been evicted yet.
2) Request during the deaf period: If the second request for d_1 arrives in the period [3, 7], then a cache miss occurs. To serve the request, d_1 has to be fetched from the backend storage.
3) Request before overhearing: If the second request for d_1 arrives in the period [7, 8], then d_1 is still a cache miss.
4) Request after overhearing: If the second request for d_1 arrives after epoch 8, then it is a cache hit and the request can be served from the cache.

Note that how to strategically choose τ_i and ω_i parameters is crucial to efficiently utilize the limited cache space. For example, if ω_i is very large, the next request for d_i is very

likely to be generated before the deaf period expires, and the hit ratio will be low. Instead, if ω_i takes a small value, we could overhear and load it into the cache at the very early stage, but it would be a waste of cache space if d_i will not be requested in the near future.

To further expand the design space, we allow randomization for the item policies. Let

$$\Pi^{\text{rco}} \triangleq \Big\{\pi^{\text{rco}}\big((q^{(1)},\cdots,q^{(n)}),(\tau^{(1)},\cdots,\tau^{(n)}),(\omega^{(1)},\cdots,\omega^{(n)})\big) : 0 \le q^{(j)} \le 1, \sum_{j=1}^{n} q^{(j)} = 1, 0 \le \tau^{(j)} \le \omega^{(j)}, 1 \le j \le n, n \in \mathbb{N}\Big\}$$

denote the set of all possible randomized item policies based on Π^{co}, where the superscript "rco" stands for randomized caching and overhearing. Each randomized item policy is a randomization of n deterministic item policies in Π^{co}, where n could be any positive integer and $q^{(j)}$ is the probability to apply the j-th deterministic item policy $\pi^{\text{co}}(\tau^{(j)},\omega^{(j)})$. Suppose d_i is served by $\pi^{\text{rco}}(\mathbf{q}_i,\boldsymbol{\tau}_i,\boldsymbol{\omega}_i)$ with $\mathbf{q}_i = (q_i^{(1)},\cdots,q_i^{(n)})$, $\boldsymbol{\tau}_i = (\tau_i^{(1)},\cdots,\tau_i^{(n)})$ and $\boldsymbol{\omega}_i = (\omega_i^{(1)},\cdots,\omega_i^{(n)})$. Every time the data item d_i is requested, a deterministic item policy $\pi^{\text{co}}(\tau_i^{(j)},\omega_i^{(j)})$ will be selected with a probability $q_i^{(j)}$ and applied to update the cache content, $1 \le j \le n$. Notably, each item d_i can be served by its customized item policy with carefully-selected parameters $\mathbf{q}_i$, $\boldsymbol{\tau}_i$ and $\boldsymbol{\omega}_i$. And the caching decisions for different data items are independent. Thus, we can analyze the hit ratio of each data item separately. Since the caches are homogeneous, we assume that the item policies for the same data item are identical on difference caches.

D. Hit Ratio Maximization

For each data item d_i, $1 \le i \le N$, we define its expected hit ratio achieved by an item policy π_i on an edge cache as

$$h_i(\pi_i) \triangleq \mathbb{E}\left[\lim_{T\to\infty} \frac{\text{Number of hits for } d_i \text{ during } [0,T] \text{ under } \pi_i}{\text{Number of requests for } d_i \text{ during } [0,T]}\right].$$

Let p_i denote the probability that a request is for data item d_i, $1 \le i \le N$, which can be calculated as

$$p_i = \frac{1}{s_i + 1/\beta_i} \Bigg/ \sum_{j=1}^{N} \frac{1}{s_j + 1/\beta_j}. \tag{2}$$

Since the demand dynamics are homogeneous across different caches, the overall expected hit ratio of all M caches is equal to the expected hit ratio of a single cache, which can be expressed by $\sum_{i=1}^{N} p_i h_i(\pi_i)$. We would like to maximize the overall expected hit ratio under the cache capacity constraint. For an edge cache, define the expected cache occupancy for d_i as

$$r_i(\pi_i) \triangleq \mathbb{E}\Big[\lim_{T\to\infty} \frac{1}{T} \cdot \big(\text{Duration when } d_i \text{ is stored in the cache during } [0,T]\big)\Big],$$

which characterizes the average cache space used for storing d_i. The cache capacity constraint states that the expected cache occupancy of all data items should not exceed the cache size. Notably, the cache capacity constraint considers the average cache occupancy rather than the real-time cache occupancy. We adopt the cache capacity constraint in an average sense for the following reason:

- It simplifies the analysis of the hit ratio maximization problem, which enables us to design efficient caching policies with provable performance.
- The caching policy obtained under the average cache capacity constraint could be easily generalized to satisfy to real-time cache capacity constraint with minor performance regressions. For example, if loading an overheard item into cache will violate the real-time cache capacity constraint, we have the option to reject the operation.

Formally, we propose the hit ratio maximization problem

$$\begin{aligned} \max_{\pi_i} \quad & \sum_{i=1}^{N} p_i \cdot h_i(\pi_i) \\ \text{subject to} \quad & \pi_i \in \Pi^{\text{rco}}, \qquad 1 \le i \le N, \\ & \sum_{i=1}^{N} r_i(\pi_i) \le b. \end{aligned} \tag{3}$$

The objective is to maximize the overall hit ratio of an edge cache by selecting the optimal item policy for each data item from the item policy set Π^{rco}. Since the demand dynamics are homogeneous across different caches, applying the optimal policy of the proposed problem to all M caches should maximize the overall hit ratio of the entire system.

Note that the optimal caching policies will not change over time, instead it captures the statistics of the demand dynamics and maximizes the expected overall hit ratio. However, if the statistics used to characterize the demand dynamics (i.e., s_i and β_i) are time varying, it would necessitate the optimal policy to change over time. Such a topic, however, falls outside the scope of this paper's discussion. Next, we will investigate this problem under the two different overhearing settings, i.e., time-driven and event-driven overhearing.

III. Edge Caching with Time-Driven Overhearing

In this section, we consider the time-driven overhearing scenario where the overhearing processes of the users are governed by independent processes. This is particularly relevant to scenarios where the base station broadcasts the items at a regular rate. In particular, we assume that the overhearing opportunity of the data item d_i is a Poisson process with rate λ_i, $1 \le i \le N$. Each cache may decide to load the overheard item into its cache or not, depending on the caching and overhearing policy.

A. Hit Ratios and Cache Occupancies

Since the caches are homogeneous and the overhearing process is independent of the number of caches, it suffices to analyze the system with a single cache. To simplify the notations, we use $h_i^{\text{co}}(\tau_i,\omega_i) \triangleq h_i(\pi^{\text{co}}(\tau_i,\omega_i))$ and

$r_i^{\rm co}(\tau_i,\omega_i) \triangleq r_i(\pi^{\rm co}(\tau_i,\omega_i))$ to denote the expected hit ratio and cache occupancy of the data item d_i, if it is served by the deterministic item policy $\pi^{\rm co}(\tau_i,\omega_i)$. In Theorem 1, we characterize $h_i^{\rm co}(\tau_i,\omega_i)$ and $r_i^{\rm co}(\tau_i,\omega_i)$ explicitly.

Theorem 1. *For time-driven overhearing, if the data item d_i is served by a deterministic item policy $\pi^{co}(\tau_i,\omega_i) \in \Pi^{co}$, we have*
(1) for $\tau_i \le \omega_i \le s_i$,

$$h_i^{co}(\tau_i,\omega_i) = 1 - \frac{\beta_i}{\lambda_i+\beta_i}\exp(-\lambda_i(s_i-\omega_i)),$$
$$r_i^{co}(\tau_i,\omega_i) = \frac{1}{\mathbb{E}[X_i]}\Big(\frac{\beta_i}{\lambda_i(\lambda_i+\beta_i)}\exp(-\lambda_i(s_i-\omega_i)) + s_i - \omega_i - \frac{1}{\lambda_i} + \frac{1}{\beta_i}\Big),$$

(2) for $\tau_i \le s_i \le \omega_i$,

$$h_i^{co}(\tau_i,\omega_i) = \frac{\lambda_i}{\lambda_i+\beta_i}\exp(-\beta_i(\omega_i-s_i)),$$
$$r_i^{co}(\tau_i,\omega_i) = \frac{1}{\mathbb{E}[X_i]}\cdot\frac{\lambda_i}{\beta_i(\lambda_i+\beta_i)}\exp(-\beta_i(\omega_i-s_i)),$$

(3) for $s_i \le \tau_i \le \omega_i$,

$$h_i^{co}(\tau_i,\omega_i) = 1 - \exp(-\beta_i(\tau_i-s_i)) + \frac{\lambda_i}{\lambda_i+\beta_i}\exp(-\beta_i(\omega_i-s_i)),$$
$$r_i^{co}(\tau_i,\omega_i) = 1 + \frac{1}{\mathbb{E}[X_i]}\cdot\Big(\frac{\lambda_i}{\beta_i(\lambda_i+\beta_i)}\exp(-\beta_i(\omega_i-s_i)) - \frac{1}{\beta_i}\exp(-\beta_i(\tau_i-s_i))\Big),$$

where X_i is defined as the inter-request time for d_i and $\mathbb{E}[X_i] = s_i + 1/\beta_i$.

The proof of Theorem 1 is presented in Appendix A. Note that for a fixed $\tau_i < s_i$, $h_i^{\rm co}$ is concave with respect to ω_i when $\omega_i < s_i - \tau_i$ and convex when $\omega_i \ge s_i - \tau_i$. Therefore, the original problem (3) is a nonconvex optimization problem, which is difficult to solve in general. However, for this specific problem, we could find the global optimum by exploiting an informative structure of the optimal solution, which will be presented in Section III-B.

Next, we will leverage Theorem 1 to calculate the expected hit ratio and cache occupancy for randomized item policies. Consider a randomized item policy $\pi_i^{\rm rco}(\mathbf{q}_i,\boldsymbol{\tau}_i,\boldsymbol{\omega}_i)$ for the data item d_i with $\mathbf{q}_i = (q_i^{(1)},\cdots,q_i^{(n)})$, $\boldsymbol{\tau}_i = (\tau_i^{(1)},\cdots,\tau_i^{(n)})$, $\boldsymbol{\omega}_i = (\omega_i^{(1)},\cdots,\omega_i^{(n)})$. Let $h_i^{\rm rco}$ and $r_i^{\rm rco}$ denote the expected hit ratio and the expected cache occupancy achieved by $\pi_i^{\rm rco}$. We derive the explicit expression for $h_i^{\rm rco}$ and $r_i^{\rm rco}$ in the following theorem.

Theorem 2. *For time-driven overhearing, if the data item d_i is served by a randomized item policy $\pi^{rco}(\mathbf{q}_i,\boldsymbol{\tau}_i,\boldsymbol{\omega}_i)$, then we have*

$$h_i^{rco}(\mathbf{q}_i,\boldsymbol{\tau}_i,\boldsymbol{\omega}_i) = \sum_{j=1}^{n} q_i^{(j)}\cdot h_i^{co}(\tau_i^{(j)},\omega_i^{(j)}),$$
$$r_i^{rco}(\mathbf{q}_i,\boldsymbol{\tau}_i,\boldsymbol{\omega}_i) = \sum_{j=1}^{n} q_i^{(j)}\cdot r_i^{co}(\tau_i^{(j)},\omega_i^{(j)}),$$

where $h_i^{co}(\tau_i^{(j)},\omega_i^{(j)})$, $r_i^{co}(\tau_i^{(j)},\omega_i^{(j)})$ can be explicitly characterized by Theorem 1.

It is shown that the expected hit ratio and cache occupancy of a randomized item policy can be calculated as the linear combination of the ones of its basic policies.

B. Informative Structure of Optimal Policies

In this section, we will prove a special structure of the optimal caching and overhearing policies, which significantly simplifies the optimization problem. Intuitively, for each data item, an item policy utilizes the cache space as the resource to achieve a high hit ratio which can be viewed as the revenue. Thus, to evaluate how efficient an item policy is, a straightforward approach is to characterize the relationship between the hit ratio and the cache occupancy achieved by it.

For each data item d_i, the hit ratio and the cache occupancy that can be achieved by some item policy in the set $\Pi^{\rm rco}$ can be described by an achievable region in a two-dimensional space. Formally, define the achievable region for the data item d_i as

$$\mathcal{R}_i^{\rm co} = \{(r,h) : \text{there exists } \pi_i \in \Pi^{\rm rco} \text{ such that } \pi_i \text{ achieves a cache occupancy } r \text{ and a hit ratio } h \text{ for } d_i\}.$$

To better characterize the achievable region $\mathcal{R}_i^{\rm co}$, we investigate two specific item policies. Define

$$\pi^{\rm c}(\tau) \triangleq \pi^{\rm co}(\tau,\infty) \quad \text{and} \quad \pi^{\rm o}(\omega) \triangleq \pi^{\rm co}(0,\omega).$$

The *caching-only* item policy $\pi^{\rm c}(\tau)$ is a specific case of $\pi^{\rm co}(\tau,\omega)$ with a caching TTL τ and an infinite overhearing TTL, i.e., never overhearing. The *overhearing-only* item policy $\pi^{\rm o}(\omega)$ is also a specific case with an overhearing TTL ω and a caching TTL zero, i.e., evicting the item immediately after serving its request.

If we restrict the item policy to be selected from the item policy set $\{\pi^{\rm c}(\tau) : \tau \ge 0\}$ or $\{\pi^{\rm o}(\omega) : \omega \ge 0\}$, then the achievable region will degenerate to a curve, since it can be parameterized in one variable (i.e., τ or ω, respectively). Therefore, the hit ratio achieved by $\pi^{\rm c}(\tau)$ and $\pi^{\rm o}(\omega)$ can be viewed as a function of the cache occupancy. Formally, for each data item d_i, define

$$h_i^{\rm c}(r) \triangleq h_i^{\rm co}(\tau_i,+\infty) \ \text{ and } \ h_i^{\rm o}(r) \triangleq h_i^{\rm co}(0,\omega_i),$$

where τ_i is selected such that $r_i^{\rm co}(\tau_i,+\infty) = r$, and ω_i is selected such that $r_i^{\rm co}(0,\omega_i) = r$. For an item d_i, $h_i^{\rm c}(r)$ (respectively, $h_i^{\rm o}(r)$) is the expected hit ratio achieved by the item policy $\pi^{\rm c}(\tau_i)$ (respectively, $\pi^{\rm o}(\omega_i)$) such that the average cache space used to store d_i is r. The $h_i^{\rm c}(r)$ and $h_i^{\rm o}(r)$ functions can be easily derived based on Theorem 1. We plot these two functions in Fig. 6.

Note that, by setting $\tau_i = +\infty$, the item policy $\pi^{\rm c}(\tau_i)$ can achieve its maximal hit ratio 1 and the maximal cache occupancy 1. Under this setting, the data item will always be stored in the cache. By setting $\omega_i = 0$, the item policy $\pi^{\rm o}(\omega_i)$ can achieve its maximal hit ratio $h_i^{\rm o}(r_i^{\rm co}(0,0)) = h_i^{\rm co}(0,0)$, which is smaller than 1. The reason is that the request of d_i may arrive before the overhearing opportunities even when we set $\omega_i = 0$. Based on Theorem 2, any points on the

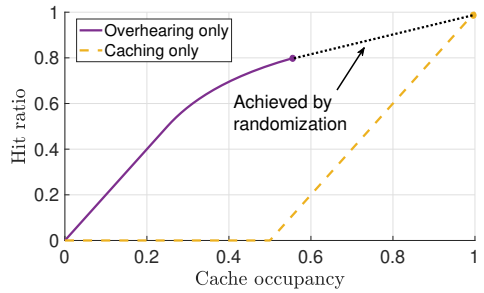


Fig. 6: Hit ratio and cache occupancy achieved by the caching-only item policies and the overhearing-only item policies with $s_i = \beta_i = \lambda_i = 1$.

line segment connecting $(1,1)$ and $(r_i^{\text{co}}(0,0), h_i^{\text{co}}(0,0))$ can be achieved by a randomization of $\pi^{\text{c}}(+\infty)$ and $\pi^{\text{o}}(0)$.

Formally, we define a randomized caching and overhearing item policy set

$$\widetilde{\Pi}^{\text{rco}} = \{\pi^{\text{o}}(\omega) : \omega \geq 0\} \cup \{\pi^{\text{rco}}((q, 1-q), (+\infty, 0), (+\infty, 0)) : 0 \leq q \leq 1\}. \quad (4)$$

The item policy set $\widetilde{\Pi}^{\text{rco}}$ contains all overhearing-only item policies and all possible randomizations of $\pi^{\text{c}}(+\infty)$ and $\pi^{\text{o}}(0)$. The achievable region of $\widetilde{\Pi}^{\text{rco}}$ can also be characterized by a curve. We define the $h_i^{\text{rco}}(r)$ function as the hit ratio achieved by an item policy from $\widetilde{\Pi}^{\text{rco}}$ when the cache occupancy is r. For $r \in [0, r^{\text{co}}(0,0)]$, we have $h_i^{\text{rco}}(r) = h_i^{\text{o}}(r)$. For $r \in (r^{\text{co}}(0,0), 1]$, $h_i^{\text{rco}}(r)$ is the line segment connecting the points $(r_i^{\text{co}}(0,0), h_i^{\text{co}}(0,0))$ and $(1,1)$. In Fig. 6, $h_i^{\text{rco}}(r)$ is the curve labeled by "overhearing only" and the line segment achieved by randomization. Notably, every point on the curve $h_i^{\text{rco}}(r)$ corresponds to exact one policy in the set $\widetilde{\Pi}^{\text{rco}}$, and vice versa. Based on Theorems 1 and 2, we can easily calculate the parameters (i.e., ω, q) for an item policy from $\widetilde{\Pi}^{\text{rco}}$ that achieves a given hit ratio or cache occupancy.

Next, we show an insightful characteristic of the achievable region $\mathcal{R}_i^{\text{co}}$.

Lemma 1. *For any $(r,h) \in \mathcal{R}_i^{co}$, we have $h \leq h_i^{rco}(r)$.*

The proof is presented in Appendix B. Lemma 1 shows that the upper boundary of the achievable region $\mathcal{R}_i^{\text{co}}$ is characterized by the function $h_i^{\text{rco}}(r)$, based on which, we can prove an informative structure of the optimal policies.

Theorem 3. *For time-driven overhearing, there must exist a caching and overhearing policy $\boldsymbol{\pi}^* = (\pi_1^*, \cdots, \pi_N^*)$ which is the optimal solution of problem (3) and satisfies $\pi_i^* \in \widetilde{\Pi}^{rco}$ for any $1 \leq i \leq N$.*

Theorem 3 is a direct application of Lemma 1. It shows that an optimal solution of problem (3) can be found from the set $\widetilde{\Pi}^{\text{rco}}$, which significantly narrows the design space. By leveraging this informative structure, we solve the optimal caching and overhearing policy in the next section.

C. Optimal Policy for Time-Driven Overhearing

Directly replacing the policy set in problem 3 with $\widetilde{\Pi}^{\text{rco}}$ will still result in a nonconvex optimization. Instead, we will solve this problem by following two steps.

Step 1: Solve the optimal r_i^*'s of the following problem (5)

$$\begin{aligned} \max_{r_i} \quad & \sum_{i=1}^{N} p_i \cdot h_i^{\text{rco}}(r_i) \quad (5) \\ \text{subject to} \quad & 0 \leq r_i \leq 1, \quad 1 \leq i \leq N, \\ & \sum_{i=1}^{N} r_i \leq b. \end{aligned}$$

Note that the original problem (3) is trying to find the optimal policy parameters (i.e., $\mathbf{q}$, $\boldsymbol{\tau}$ and $\boldsymbol{\omega}$). In Step 1, the original optimization problem in the domain of policy parameters is converted into a new problem in the domain of the cache occupancies (i.e., $\boldsymbol{r}$). Recall that $h_i^{\text{rco}}(r_i)$ captures the relationship between the hit ratio and the cache occupancy for item policies in the set $\widetilde{\Pi}^{\text{rco}}$. Using Theorem 3, we can prove that the optimal occupancies found by problem (5) are actually the occupancies achieved by the optimal item policies of the original problem (3). More importantly, Theorems 1 and 2 indicate that the $h_i^{\text{rco}}(r)$ functions are concave, $1 \leq i \leq N$. Therefore, the optimization problem (5) is convex and can be solved using standard tools (e.g., KKT conditions and the water-filling algorithm [42]).

Step 2: Once the optimal solution r_i^*'s of Step 1 is solved, then based on Theorems 1 and 2, we can easily find the item policies from the set $\widetilde{\Pi}^{\text{rco}}$ that achieve r_i^*'s. And these item policies form an optimal solution of the original hit ratio maximization problem (3). We use ω_i^*'s and q_i^*'s to denote the parameters for these item policies. An optimal policy for time-driven overhearing is formally proposed as follows.

Caching and overhearing policy for time-driven overhearing ($\boldsymbol{\pi}^T$): Serve each data item d_i, $1 \leq i \leq N$, by a randomized item policy $\pi^{\text{rco}}(\mathbf{q}_i, \boldsymbol{\tau}_i, \boldsymbol{\omega}_i)$ where $\mathbf{q}_i = (q_i^*, 1-q_i^*)$, $\boldsymbol{\tau}_i = (+\infty, 0)$ and $\boldsymbol{\omega}_i = (+\infty, \omega_i^*)$.

The proposed optimal policy reveals the following insights:

1) We should either evict a data item immediately after serving a request for it (i.e., set $\tau_i = 0$) or always store it in the cache (i.e, set $\tau_i = +\infty$). Setting $\tau_i \in (0, +\infty)$ will be suboptimal.
2) If we decide to bring an item back into the cache by overhearing (i.e., set $\omega_i < +\infty$), then that item should be evicted immediately after serving each request for it (i.e., set $\tau_i = 0$).

These insights will guide us in the more complex scenario of event-driven overhearing, which is tackled next.

IV. Edge Caching with Event-Driven Overhearing

In this section, we consider a more realistic overhearing setting. When a cache miss happens, the data item will be fetched from the backend and sent to the user over broadcast channels. Meanwhile, the other caches could overhear and decide whether they would like to store this data item. Since the overhearing opportunities are triggered by cache misses, the event-driven overhearing process depends on the caching decisions as well as the number of caches in the system. It can be easily verified that the overhearing process are not Poisson under this setting. As a result, the analysis for time-driven

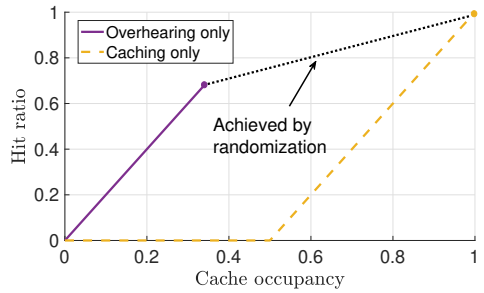


Fig. 7: Hit ratio and cache occupancy achieved by the caching-only item policies and the overhearing-only item policies with $s_i = \beta_i = 1$ and $M = 10$.

overhearing cannot be directly applied for the event-driven scenario.

A. Hit Ratios and Cache Occupancies

It is difficult to derive the hit ratio and the cache occupancy for a general caching and overhearing policy, since the overhearing process is not tractable under this setting. However, we are able to characterize a few key properties for some specific policies, which inspires us to design a provably good policy.

Similar to the notations in Section III, we still use $h_i^{\mathrm{o}}(\cdot)$, $h_i^{\mathrm{c}}(\cdot)$, $h_i^{\mathrm{co}}(\cdot)$, $h_i^{\mathrm{rco}}(\cdot)$ to denote the hit ratios achieved by the item policies π^{o}, π^{c}, π^{co}, π^{rco} for the data item d_i, respectively. The same rules will also be applied to the notations for cache occupancies. However, the expression of these functions will be different from those in Section III, since the overhearing processes have been changed.

Lemma 2. *Consider the event-driven overhearing. If d_i is served by the item policy $\pi^o(\omega_i)$, then we have*

$$h_i^o(r) = (\beta_i s_i + 1)r$$

for $0 \le r \le r_i^o(s_i)$. If d_i is served by the item policy $\pi^c(\tau_i)$, then its hit ratio and occupancy are exactly the same as those achieved by $\pi^c(\tau_i)$ for time-driven overhearing, and can be directly calculated using Theorem 1.

In Lemma 2, we analyze the item policies $\pi^{\mathrm{c}}(\tau_i)$ and $\pi^{\mathrm{o}}(\omega_i)$ under event-driven overhearing. The proof is presented in Appendix C. For the caching-only item policy $\pi^{\mathrm{c}}(\tau_i)$, the hit ratio and the cache occupancy are exactly the same as the ones under time-driven overhearing, since $\pi^{\mathrm{c}}(\tau_i)$ sets $\omega_i = +\infty$ and is independent of the overhearing process. For the overhearing-only item policy $\pi^{\mathrm{o}}(\omega_i)$, the hit ratio of d_i is a linear function with respect to the cache occupancy when $0 \le r \le r^{\mathrm{o}}(s_i) = r^{\mathrm{co}}(0, s_i)$, or, equivalently when $\omega_i \ge s_i$. When $\omega_i < s_i$, the overhearing-only item policy becomes intractable. We plot the hit ratio and the cache occupancy that can be achieved by $\pi^{\mathrm{o}}(\omega_i)$ with $\omega_i \ge s_i$ and $\pi^{\mathrm{c}}(\tau_i)$ with $\tau_i \ge 0$ in Fig. 7.

In Lemma 2 we characterize the relationship between hit ratios and cache occupancies for event-driven overhearing, but we are not able to analytically solve the parameter ω_i that achieves a given cache occupancy r. To address this issue, we first assume that the policy parameter ω_i to achieve any $r \le r^{\mathrm{co}}(0, s_i)$ is solvable. With this assumption, we will propose provably good policies in Section IV-B. Then, in Section IV-C, we will discuss how to implement these policies in real practice without the proposed assumption.

B. Asymptotically Optimal Policy for Event-Driven Overhearing

Although the hit ratio and the cache occupancy under event-driven overhearing are not fully tractable, we could still design provably good polices by leveraging the insights obtained from the optimal structure under time-driven overhearing. The general idea is to first construct a shrunken policy set, which contains less item policies but retains some tractability under event-driven overhearing. Then, we will find the best policy from the shrunken policy set and analytically characterize its performance.

For each data item d_i, $1 \le i \le N$, define a policy set

$$\widehat{\Pi}_i^{\mathrm{rco}} = \{\pi^{\mathrm{o}}(\omega_i) : \omega_i \ge s_i\} \cup \{\pi^{\mathrm{rco}}((q_i, 1-q_i), (+\infty, 0), (+\infty, s_i)) : 0 \le q_i \le 1\}. \quad (6)$$

The set $\widehat{\Pi}_i^{\mathrm{rco}}$ contains overhearing-only item policies $\pi^{\mathrm{o}}(\omega_i)$ with $\omega_i \ge s_i$ and all possible randomizations of $\pi^{\mathrm{o}}(s_i)$ and $\pi^{\mathrm{c}}(+\infty)$. The reason to construct such policy sets is that we could characterize the relationship between hit ratios and cache occupancies for these item policies based on Lemma 2. Instead of solving the original problem (3), we would like to find the best policy from the shrunken policy sets by solving the following problem

$$\begin{aligned} \max_{\pi_i} \quad & \sum_{i=1}^{N} p_i \cdot h_i(\pi_i) \\ \text{subject to} \quad & \pi_i \in \widehat{\Pi}_i^{\mathrm{rco}}, \qquad 1 \le i \le N, \\ & \sum_{i=1}^{N} r_i(\pi_i) \le b. \end{aligned} \quad (7)$$

Define $\widehat{h}_i^{\mathrm{rco}}(r)$ as the hit ratio of d_i achieved by an item policy from the set $\widehat{\Pi}_i^{\mathrm{rco}}$ such that the cache occupancy is r. We have $\widehat{h}_i^{\mathrm{rco}}(r) = (s_i\beta_i + 1)r$ for $0 \le r \le r_i^{\mathrm{co}}(0, s_i)$, and

$$\widehat{h}_i^{\mathrm{rco}}(r) = \frac{1 - (s_i\beta_i + 1)r_i^{\mathrm{co}}(0, s_i)}{1 - r_i^{\mathrm{co}}(0, s_i)} \cdot r + \frac{s_i\beta_i r_i^{\mathrm{co}}(0, s_i)}{1 - r_i^{\mathrm{co}}(0, s_i)}$$

for $r_i^{\mathrm{co}}(0, s_i) \le r \le 1$. The $\widehat{h}_i^{\mathrm{rco}}(r)$ curve is illustrated in Fig. 7 as the line segments achieved by overhearing only and randomization. Every point on the curve $\widehat{h}_i^{\mathrm{rco}}(r)$ corresponds an item policy in the set $\widehat{\Pi}_i^{\mathrm{rco}}$, and vice versa. To find the best item policies from $\widehat{\Pi}_i^{\mathrm{rco}}$, we formulate the following optimization problem.

$$\begin{aligned} \max_{r_i} \quad & \sum_{i=1}^{N} p_i \cdot \widehat{h}_i^{\mathrm{rco}}(r_i) \\ \text{subject to} \quad & 0 \le r_i \le 1, \quad 1 \le i \le N, \\ & \sum_{i=1}^{N} r_i \le b. \end{aligned} \quad (8)$$

Since $\widehat{h}_i^{\text{rco}}(r)$ functions are concave, problem (8) is convex and can be solved by the same approach that solves problem (5). Let r_i^*'s denote the optimal solution to problem (8). We can easily identify the item policy from the set $\widehat{\Pi}_i^{\text{rco}}$ that achieves r_i^*, $1 \leq i \leq N$. We propose a caching and overhearing policy as follows.

Caching and overhearing policy for event-driven overhearing ($\boldsymbol{\pi}^E$): Let each data item d_i, $1 \leq i \leq N$, be served by the item policy from the set $\widehat{\Pi}_i^{\text{rco}}$ that achieves the cache occupancy r_i^*, i.e., the solution of (8).

To analytically characterize the performance of the proposed policy $\boldsymbol{\pi}^E$, we first introduce an upper bound for the achievable hit ratio. For a system consisting of M caches, let $h^*(M)$ denote the overall hit ratio achieved by the optimal solution of problem (3) under event-driven overhearing. In the following lemma, we prove that $h^*(M)$ is upper bounded by a constant which is defined as h^{upper}.

Lemma 3. *Consider a system of M caches where each cache has a size b. We have*

$$h^*(M) \leq \sum_{i=1}^{K} p_i + p_{K+1}(\beta_{K+1}s_{K+1}+1)\Big(b - \sum_{i=1}^{K}\frac{1}{\beta_i s_i + 1}\Big) \triangleq h^{upper}, \tag{9}$$

where K is the integer such that

$$\sum_{i=1}^{K}\frac{1}{\beta_i s_i+1} \leq b < \sum_{i=1}^{K+1}\frac{1}{\beta_i s_i+1}. \tag{10}$$

The proof of Lemma 3 is presented in Appendix D. The upper bound proposed in this lemma is actually the hit ratio achieved by an idealized policy. The idealized policy assumes that we could always overhear any data item at any time and attempts to find the best overhearing time based on the anticipated arrival time for the next request.

Let $h^E(M)$ denote the expected overall hit ratio achieved by $\boldsymbol{\pi}^E$ in a system consisting of M caches. We characterize the distance between $h^E(M)$ and h^{upper} in the following theorem.

Theorem 4. *For the proposed policy $\boldsymbol{\pi}^E$ and K defined in Equation (10), we have, as the number of caches $M \to +\infty$,*

$$0 \leq h^*(M) - h^E(M) \leq h^{upper} - h^E(M) \leq \max_{1\leq i\leq K+1} 2\sqrt{(\beta_i s_i+1)/M},$$

which implies that

$$\lim_{M\to+\infty} h^E(M) = \lim_{M\to+\infty} h^*(M) = h^{upper}.$$

In Appendix E, we prove Theorem 4 by showing that the expected inter-overhearing time for each date item will converge to zero as M goes to infinity. Theorem 4 tells us that the proposed policy $\boldsymbol{\pi}^E$ for event-driven overhearing setting is asymptotically optimal as the number of caches goes to infinity. Intuitively, as the number of caches in the system increases, it will be more likely to overhear a data item. The proposed policy could efficiently utilize the overhearing opportunities and achieve asymptotically optimal performance.

C. Discussion on Implementation

Since the overhearing process is difficult to analyze, in order to design provably good policies, we previously assumed in Section IV-A that the cache occupancy achieved by the item policy $\pi^{\text{o}}(\omega_i)$, $\omega_i \geq s_i$, can be analytically solved. Based on this tractability assumption, we propose and analyze $\boldsymbol{\pi}^E$ in Section IV-B. In this section, we will discuss how to implement the proposed policies without this assumption.

First, we note that it is impractical to estimate the cache occupancy for all possible $\pi^{\text{o}}(\omega_i)$'s, since ω_i can take any real numbers. In contrast, we will show that a good performance can be guaranteed by leveraging an accurate estimation of the cache occupancy achieved by a specific item policy $\pi^{\text{o}}(s_i)$.

For the convex problem (8), KKT conditions show that the optimal solution satisfies that

$$\sum_{i=1}^{N}\mathbf{1}(r_i^* \neq r_i^{\text{o}}(s_i) \text{ and } 0 < r_i^* < 1) \leq 1, \tag{11}$$

where $r_i^{\text{o}}(s_i)$ is the cache occupancy achieved by the overhearing-only item policy $\pi^{\text{o}}(\omega_i)$ with $\omega_i = s_i$. It indicates that there is at most one r_i^* that takes a value other than $r_i^{\text{o}}(s_i)$, 0 and 1. In other word, except for one item policy, the other item policies in the optimal solution must be the overhearing-only item policy $\pi^{\text{o}}(s_i)$ or a randomization of $\pi^{\text{o}}(s_i)$ and $\pi^{\text{c}}(+\infty)$. As a result, we could further narrow down the policy set by considering $\pi^{\text{o}}(s_i)$ and randomizations of $\pi^{\text{o}}(s_i)$ and $\pi^{\text{c}}(+\infty)$, i.e.,

$$\{\pi^{\text{rco}}((q_i, 1-q_i),(+\infty,0),(+\infty,s_i)) : 0 \leq q_i \leq 1\}, \tag{12}$$

Once we have a good estimation of $r_i^{\text{o}}(s_i)$, all item policies in this set are tractable.

Therefore, to implement the caching and overhearing policy $\boldsymbol{\pi}^E$ proposed in Section IV-B, we could solve the hit ratio maximization problem based on the policy set (12) rather than the one defined in (6). The solved policy is an approximation of $\boldsymbol{\pi}^E$. By applying (11), we can prove that the overall hit ratio achieved by this approximated policy is within $1 - 1/b$ fraction of $\boldsymbol{\pi}^E$, where b is the cache size.

The remaining problem is how to estimate $r_i^{\text{o}}(s_i)$ values. A simple idea is to first run an estimation phase to approximate $r_i^{\text{o}}(s_i)$ and then solve the modified policies using the estimated values. In the estimation phase, for each data item d_i, we may run $\pi^{\text{o}}(s_i)$ for T units of time, and estimate $r_i^{\text{o}}(s_i)$ by

$$\bar{r}_i^{\text{o}}(s_i) = (\text{Duration of time when } d_i \text{ is cached})\,/T.$$

The proposed implementation solution could introduce performance losses compared to the original policy $\boldsymbol{\pi}^E$ due to the following two reasons:

1) The implemented policy is an approximation of $\boldsymbol{\pi}^E$ by considering the policy set (12) rather than (6).
2) The caching and overhearing decisions are biased since the estimation $\bar{r}_i^{\text{o}}(s_i)$ is not accurate.

However, these performance losses could be ignored as long as the cache size b and the length of the estimation phase T are sufficiently large.

V. Generalization for Heterogeneous Demand Dynamics

In the main paper, we focus on the homogeneous demand dynamics, where different users have the same demand pattern (i.e., s_i, β_i) for a given data item d_i. The obtained insights and theorems can be easily generalized for heterogeneous demands with minor modifications.

We use m, $1 \le m \le M$, to index an edge cache or the user served by the corresponding edge cache interchangeably, since each edge cache is assumed to serve a single user. Our first step is to extend the proposed ON-OFF processes to allow for different demand patterns of the same data item among different users. For the user m, we use the proposed ON-OFF process with a OFF-period length $s_i^{(m)}$ and ON-period request rate $\beta_i^{(m)}$ to describe the demand dynamics of the data item d_i. The popularity of d_i for user m can be evaluated by

$$p_i^{(m)} = \left(s_i^{(m)} + 1/\beta_i^{(m)}\right)^{-1} \Big/ \sum_{j=1}^{N} \left(s_j^{(m)} + 1/\beta_j^{(m)}\right)^{-1}.$$

Define $\nu^{(m)}$ as

$$\nu^{(m)} = \sum_{i=1}^{N} \left(s_i^{(m)} + 1/\beta_i^{(m)}\right)^{-1} \Big/ \sum_{m=1}^{M}\sum_{i=1}^{N} \left(s_i^{(m)} + 1/\beta_i^{(m)}\right)^{-1}.$$

$\nu^{(m)}$ represents the ratio of requests that are from user m. We have $\sum_{m=1}^{M} \nu^{(m)} = 1$.

Similar to the hit ratio and the cache occupancy defined in Section II-D, we use $h_i^{(m)}(\pi)$ and $r_i^{(m)}(\pi)$ to denote the hit ratio and the cache occupancy of the data item d_i in the edge cache m achieved by the item policy π, respectively.

Our goal is to find the optimal policy such that the overall hit ratio of the entire system is maximized. We formally propose the problem as follows.

$$\begin{aligned} \max_{\pi_i^{(m)}, 1\le m\le M, 1\le i\le N} \quad & \sum_{m=1}^{M}\sum_{i=1}^{N} \nu^{(m)} p_i^{(m)} \cdot h_i^{(m)}\left(\pi_i^{(m)}\right) \quad (13) \\ \text{subject to} \quad & \pi_i^{(m)} \in \Pi^{\text{rco}}, 1 \le m \le M, 1 \le i \le N, \\ & \sum_{i=1}^{N} r_i^{(m)}\left(\pi_i^{(m)}\right) \le b, \ \ 1 \le m \le M. \end{aligned}$$

A. Time-Driven Overhearing

The time-driven overhearing process is independent of the edge caching policy. Under time-driven overhearing, the cache hit ratio and occupancy of an edge cache are determined by its own policy and are independent of other caches. Therefore, maximizing the overall hit ratio of the entire system is equivalent to maximizing the hit ratio of each edge cache separately. Formally, we can propose M sub-problems, where the m-th problem is defined as follows.

$$\begin{aligned} \max_{\pi_i^{(m)}, 1\le i\le N} \quad & \nu^{(m)} p_i^{(m)} \cdot \sum_{i=1}^{N} h_i^{(m)}\left(\pi_i^{(m)}\right) \quad (14) \\ \text{subject to} \quad & \pi_i^{(m)} \in \Pi^{\text{rco}}, \ \ 1 \le i \le N, \\ & \sum_{i=1}^{N} r_i^{(m)}\left(\pi_i^{(m)}\right) \le b. \end{aligned}$$

Let $\{\pi_i^{(m)*} : 1 \le i \le N\}$ denote the optimal solution of the sub-problem (14), then $\{\pi_i^{(m)*} : 1 \le i \le N, 1 \le m \le M\}$ would be the optimal solution of the original problem (13). Notably, solving the sub-problem (14) is equivalent to solving problem (3) that was proposed for the homogeneous setting previously. Therefore, our analysis for the homogeneous demand setting in Section III is still valid for the heterogeneous setting with time-driven overhearing. And the proposed $\boldsymbol{\pi}^T$ policy would be the solution to the sub-problem (14), where the policy inputs s_i, β_i are replaced by $s_i^{(m)}$ and $\beta_i^{(m)}$.

B. Event-Driven Overhearing

The event-driven overhearing process becomes more complicated when the demand dynamics are heterogeneous across different edge caches. We are not able to split the original overall hit ratio maximization problem to independent sub-problems designed for each cache, since the policy for one cache will impact the overhearing processes as well as the optimal decision of other caches.

Due to the increased complexity, the policy $\boldsymbol{\pi}^E$ proposed for homogeneous demand dynamics cannot be directly extended for heterogeneous settings. Fortunately, the key properties of hit ratios and cache occupancies characterized in Lemma 2 are still valid for each edge cache. In particular, for the edge cache m, Lemma 2 holds if we replace s_i, β_i by $s_i^{(m)}$, $\beta_i^{(m)}$. Similar to the idea of proposing policy $\boldsymbol{\pi}^E$ for homogeneous demands in Section IV, we will leverage Lemma 2 and the informative structure characterized in Section III-B to design a provably good policy.

For the cache m and the data item d_i, let $\zeta(i,m)$ be a reordering of the data index, such that $\zeta(i,m)$ takes distinct integer values in $[1,N]$ for different input i, and $\beta_{\zeta(i,m)}^{(m)} \ge \beta_{\zeta(j,m)}^{(m)}$ for any $1 \le i < j \le N$. Define the set of items $\mathcal{D}^{(m)} = \{d_i : 1 \le \zeta(i,m) \le K^{(m)}\}$, where $K^{(m)}$ is an integer such that

$$\sum_{i=1}^{K^{(m)}} \frac{1}{\beta_{\zeta(i,m)}^{(m)} s_i^{(m)} + 1} \le b < \sum_{i=1}^{K^{(m)}+1} \frac{1}{\beta_{\zeta(i,m)}^{(m)} s_i^{(m)} + 1}. \quad (15)$$

We say the data item d_i is a popular data item for user m, if $\zeta(i,m) \le K^{(m)}$. Define $\mathcal{C}_i = \{m : \zeta(i,m) \le K^{(m)}\}$ to be the set of users, for which d_i is a popular data item. We propose the policy $\boldsymbol{\pi}^{E-O}$ as follows.

Overhearing only policy for event-driven overhearing ($\boldsymbol{\pi}^{E-O}$): For each edge cache m, apply the overhearing only item policy $\pi^{\text{o}}(s_i^{(m)})$ to serve the data item d_i, if $d_i \in \mathcal{D}^{(m)}$. Do not overhear or cache the data items that are not in $\mathcal{D}^{(m)}$.

$\boldsymbol{\pi}^{E-O}$ imitates the overhearing decisions of $\boldsymbol{\pi}^E$, which is proposed for homogeneous demand dynamics in Section IV-B. However, to simplify the analysis, $\boldsymbol{\pi}^{E-O}$ forgoes the caching only options of $\boldsymbol{\pi}^E$. For M caches under event-driven overhearing, let $h^*(M)$ and $h^{E-O}(M)$ denote the overall expected hit ratio achieved by the optimal policy and the proposed policy $\boldsymbol{\pi}^{E-O}$, respectively. Assume that $\beta_i^{(m)}$ is upper bounded and define $\beta_{max} = \max_{1\le i\le N, 1\le m\le M} \beta_i^{(m)}$. We can prove that $h^{E-O}(M)$ is close to $h^*(M)$.

Theorem 5. *For the proposed policy $\boldsymbol{\pi}^{E-O}$, we have*

$$0 \leq h^*(M) - h^{E-O}(M) \leq \frac{1}{b} + \max_{1\leq i\leq N} \frac{2\sqrt{\beta_{max}}}{\sqrt{\sum_{m\in\mathcal{C}_i}\left(s_i^{(m)} + 1/\beta_i^{(m)}\right)^{-1}}}.$$

The proof of this theorem uses a similar approach that proves Theorem 4, and therefore is omitted due to the page limit. The detailed proof can be found in the technical report [43]. Theorem 5 indicates $\lim_{M\to+\infty} h^*(M) - h^{E-O}(M) = 1/b$, if we have, for any $1 \leq i \leq N$

$$\lim_{M\to+\infty} \sum_{m\in\mathcal{C}_i} \left(s_i^{(m)} + 1/\beta_i^{(m)}\right)^{-1} \to +\infty. \tag{16}$$

Condition (16) states that as the user population grows, the overall request rate for d_i from the user set $\mathcal{C}_i$ also keeps increasing. Under such conditions, we would have more and more overhearing opportunities as the user population increases. Theorem 5 reveals an insight that near optimal caching performance can be achieved by strategically leveraging the overhearing opportunities (i.e., $\boldsymbol{\pi}^{E-O}$), if the demand of each data item increases consistently when it is exposed to a larger user population.

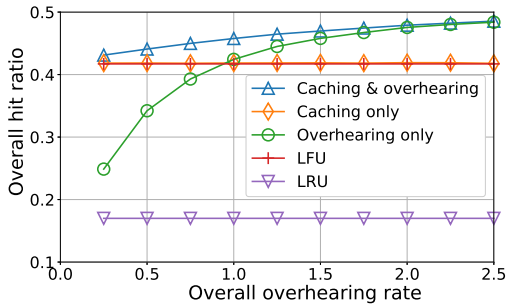
Unlike the policy $\boldsymbol{\pi}^{E}$ that can achieve asymptotically optimal performance for homogeneous demand settings, the policy $\boldsymbol{\pi}^{E-O}$ designed for heterogeneous settings always has a $1/b$ hit ratio gap with the optimal policy. The reason is that $\boldsymbol{\pi}^{E-O}$ adopts an overhearing only mechanism and restricts the overhearing TTL $\omega_i^{(m)}$ to take either value $s_i^{(m)}$ or $+\infty$ for the ease of analysis. Considering that the cache size b is typically large, $\boldsymbol{\pi}^{E-O}$ achieves reasonably good performance.

VI. Evaluation

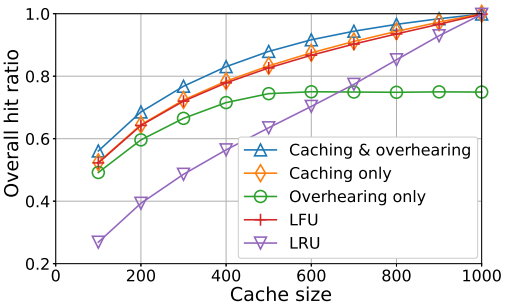
We will validate the theoretical results by evaluating the empirical performance of the proposed $\boldsymbol{\pi}^{T}$ and $\boldsymbol{\pi}^{E}$ policies and comparing them with the following benchmarks:

- The optimal overhearing-only policy: this policy is the solution of problem (3) with an additional constraint $\tau_i = 0$, $1 \leq i \leq N$, and can be easily solved using the same approach that solves $\boldsymbol{\pi}^{T}$ and $\boldsymbol{\pi}^{E}$. The optimal overhearing-only policy evicts the data item immediately after a request for it.
- The optimal caching-only policy: this policy is the solution of problem (3) with an additional constraint $\omega_i = +\infty$, $1 \leq i \leq N$, and can be easily solved using standard convex optimization tools. It turns out that the optimal caching-only policy caches data items with the largest long term popularities. and achieves better performance than various conventional policies (e.g., LRU, LFU) under the setting of this paper.
- LFU policy: the policy caches the most frequently used data items, which is an approximation of the optimal caching-only policy.
- LRU policy: the policy caches the most recently used data items.

In Experiments 1 and 2, we evaluate the performance of these policies under time-driven and event-driven overhearing settings, respectively.



(a)



(b)

Fig. 8: Overall hit ratio with time-driven overhearing.

Experiment 1: In this experiment, we consider the time-driven overhearing setting. Set $b = 50$, $N = 1000$, $\beta_i = c\cdot i^{-0.8}$ with $c = 1/\sum_{i=1}^{N} i^{-0.8}$ and $s_i = 1/\beta_i$. Let $\lambda_i = \gamma\cdot\beta_i$, where $\gamma = \sum_{i=1}^{N}\lambda_i$ is the overall overhearing rate. Since the number of caches M does not impact the performance for time-driven overhearing, we simply set $M = 1$. We evaluate the overall hit ratios under different γ values and depict the results in Figure 8a. It can be observed that the proposed optimal caching and overhearing policy $\boldsymbol{\pi}^{T}$ always outperforms the other benchmarks. The overhearing-only policy achieves similar performance as $\boldsymbol{\pi}^{T}$ when γ is large, which validates that when there are sufficient overhearing opportunities, the overhearing-only policy can achieve near optimal performance. However, when γ is small, the overhearing-only suffers a lot. The overall hit ratios achieved by the caching-only policy, LFU and LRU are constant, since they are independent of the overhearing process. LFU achieves similar performance as the caching-only policy. However, LRU achieves much worse performance, since the most recently used data item may not be popular in the near future due to the individualized demand dynamics.

Next, we fix $\gamma = 1$ and change the cache size b. The results are plotted in Fig. 8b. The caching and overhearing policy $\boldsymbol{\pi}^{T}$ still outperforms the other benchmarks. As for the overhearing-only policy, the hit ratio will be a constant less than 1, when the cache size is larger than a threshold. The reason is that the overall overhearing rate is too low, and cache is not full even when we maximize the overhearing utilization (i.e., set $\omega_i = 0$). As a result, further increasing the cache size will not lead to a higher hit ratio. The caching-only policy, LFU and LRU can achieve a hit ratio 1, when the cache is large enough to store all items (i.e., $b = N = 1000$).

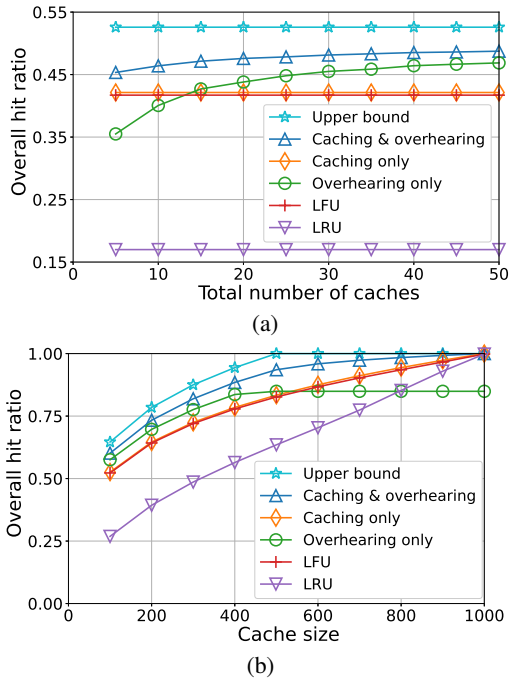


Fig. 9: Overall hit ratio with event-driven overhearing.

Experiment 2: In this experiment, we simulate the event-driven overhearing. Consider $N = 1000$ data items with $\beta_i = c \cdot i^{-0.8}$, $c = 1/\sum_{i=1}^{N} i^{-0.8}$ and $s_i = 1/\beta_i$ for $1 \leq i \leq N$. We evaluate the overall hit ratios achieved by the proposed overhearing and caching policy $\boldsymbol{\pi}^E$ as well as other benchmarks, and compare them with the upper bound of the optimal hit ratio derived in Equation (9). Note that the policy $\boldsymbol{\pi}^E$ is solved using an estimation phase with duration 10000 based on discussions in Section IV-C.

First, we set the cache size $b = 50$ and evaluate the hit ratios for different numbers of caches. The results are plotted in Figure 9a. The proposed caching and overhearing policy always achieves the best performance. When M is small, the overhearing-only policy achieves a much lower hit ratio than the caching-only policy. When M is large, the overhearing-only policy outperforms the caching-only policy. In addition, the hit ratios achieved by the caching and overhearing policy are getting closer to the upper bound h^{upper} as M increases, which validates the asymptotic optimality. LFU achieves similar performance to the optimal caching-only policy and LRU achieves the worst performance.

Next, we set $M = 50$ and evaluate the hit ratios for different cache sizes. The results are presented in Figure 9b. The proposed caching and overhearing policy $\boldsymbol{\pi}^E$ always achieves the best performance. When the cache size is less than 500, the overhearing-only policy outperforms the caching-only policy, because M is relatively large to generate sufficient overhearing opportunities. However, for $b > 500$, when we further increase the cache size, the overhearing-only policy cannot achieve a larger hit ratio. At this time, the overhearing opportunities become the bottleneck of the system, and the cache space cannot be fully utilized due to the lack of overhearing opportunities. In contrast, the proposed caching and overhearing policy, the caching-only policy, LFU and LRU can always benefit from a larger cache size.

VII. Conclusion

Edge caching typically serves a very small group of users with individualized data demand. Hence, caching schemes for an edge need to be substantially different from those at the core that serves a large population of users. In this paper, we developed new caching policies optimized for individualized data demand at the wireless edges. With the objective to maximize the overall hit ratio, we proposed to actively evict the data items that are not likely to be requested in the near future and bring them back into the cache through overhearing when they become popular again. In particular, when the overhearing opportunities are time-driven, the optimization problem turns out to be non-convex. Nevertheless, by exploiting an informative structure of the optimal solution, we converted the original problem to a convex one and found the optimal policy $\boldsymbol{\pi}^T$. When the overhearing opportunities are event-driven, the overhearing processes become intractable. Still, inspired by the optimality structure of the time-driving overhearing setting, we proposed a caching and overhearing policy $\boldsymbol{\pi}^E$ which is asymptotically optimal as the total number of caches increases. Both theoretical and numerical results verified that the caching policies designed specifically for edges could substantially improve the caching efficiency and outperform the policies designed for the core.

Appendix A
Proof of Theorem 1

Let $\pi^{\text{c}}(\tau_i) = \pi^{\text{co}}(\tau_i, +\infty)$ denote a caching-only policy that never overhears. Let $h_i^{\text{c}}(\tau_i)$ and $r_i^{\text{c}}(\tau_i)$ denote the expected hit ratio and cache occupancy of the data item d_i when it is served by the policy $\pi^{\text{c}}(\tau_i)$. Similarly, we can define $\pi^{\text{o}}(\omega_i) = \pi^{\text{co}}(0, \omega_i)$, $h_i^{\text{o}}(\omega_i)$ and $h_i^{\text{o}}(\omega_i)$. We will establish the following lemma before proving Theorem 1.

Lemma 4. *The expected hit ratio and cache occupancy achieved by $\pi^{co}(\tau_i, \omega_i)$ with $\tau_i \leq \omega_i$ can be calculated by $h^{co}(\tau_i, \omega_i) = h_i^c(\tau_i) + h_i^o(\omega_i)$ and $r^{co}(\tau_i, \omega_i) = r_i^c(\tau_i) + r_i^o(\omega_i)$.*

Proof of Lemma 4. Assume there is a request for the data item d_i at time 0 and the next request for d_i will arrive at time $\sigma > s_i$. We will first analyze the probability of having a cache hit of d_i at time σ, as well as the expected time when d_i is stored in the cache in the time interval $[0, \sigma]$.

$$\begin{aligned}
&\mathbb{P}[\text{cache hit at time } \sigma |\ d_i \text{ is served by } \pi^{\text{co}}(\tau_i, \omega_i)] \\
&\quad = \mathbb{P}[\sigma > s_i + \omega_i \text{ and } d_i \text{ is overheard during } [\omega_i, \sigma]] \\
&\qquad + \mathbb{P}[\sigma \leq s_i + \tau_i] \\
&\mathbb{P}[\text{cache hit at time } \sigma |\ d_i \text{ is served by } \pi^{\text{c}}(\tau_i)] \\
&\quad = \mathbb{P}[\sigma \leq s_i + \tau_i] \\
&\mathbb{P}[\text{cache hit at time } \sigma |\ d_i \text{ is served by } \pi^{\text{o}}(\omega_i)] \\
&\quad = \mathbb{P}[\sigma > s_i + \omega_i \text{ and } d_i \text{ is overheard during } [\omega_i, \sigma]].
\end{aligned}$$

Hence, we have

$$\begin{aligned}
&\mathbb{P}[\text{cache hit at time } \sigma |\ d_i \text{ is served by } \pi^{\text{co}}(\tau_i, \omega_i)] \\
&\quad = \mathbb{P}[\text{cache hit at time } \sigma |\ d_i \text{ is served by } \pi^{\text{c}}(\tau_i)] \\
&\qquad + \mathbb{P}[\text{cache hit at time } \sigma |\ d_i \text{ is served by } \pi^{\text{o}}(\omega_i)],
\end{aligned}$$

which indicates $h^{\text{co}}(\tau_i,\omega_i)=h_i^{\text{c}}(\tau_i)+h_i^{\text{o}}(\omega_i)$, since the policy is renewed after each data request.

Let T^{co}, T^{c} and T^{o} denote the amount of time when d_i is stored in the cache during $[0,\sigma]$, if $\pi^{\text{co}}(\tau_i,\omega_i)$, $\pi^{\text{c}}(\tau_i)$ and $\pi^{\text{o}}(\omega_i)$ are applied, respectively. We have $T^{\text{co}}=T^{\text{c}}+T^{\text{o}}$, which indicates $r^{\text{co}}(\tau_i,\omega_i)=r_i^{\text{c}}(\tau_i)+r_i^{\text{o}}(\omega_i)$. □

Proof of Theorem 1. In order to prove Theorem 1, we will derive the expected hit ratio and cache occupancy achieved by $\pi^{\text{c}}(\tau_i),\pi^{\text{o}}(\omega_i)$ in different parameter regions.

Case 1: $\tau_i\le\omega_i\le s_i$

In this case, the cached data item d_i is evicted during the OFF period. Thus, the caching-only policy $\pi^{\text{c}}(\tau_i)$ always achieves a 0 hit ratio and cache occupancy. For the overhearing-only policy $\pi^{\text{o}}(\omega_i)$, without loss of generality, we assume that the most recent request arrives at time 0. We will analyze the probability that the next request for d_i is a hit, and its expected cache occupancy. Let X_i denote the time when the next request for d_i arrives, and Y_i denote the time when we overhear d_i for the first time after the deaf period. We have

$$\begin{aligned}h_i^{\text{o}}(\omega_i)&=\mathbb{P}[\text{The next request for } d_i \text{ is a hit under } \pi^{\text{o}}(\omega_i)]\\&=\mathbb{P}[Y_i+\omega_i\le X_i]\\&=\mathbb{P}[Y_i+\omega_i\le s_i]+\mathbb{P}[s_i<Y_i+\omega_i\le X_i]\\&=1-\exp\left(-\lambda_i(s_i-\omega_i)\right)\\&\quad+\exp\left(-\lambda_i(s_i-\omega_i)\right)\cdot\frac{\lambda_i}{\lambda_i+\beta_i}\\&=1-\exp\left(-\lambda_i(s_i-\omega_i)\right)\cdot\frac{\beta}{\lambda_i+\beta_i}\end{aligned}$$

The cache occupancy of $\pi^{\text{o}}(\omega_i)$ is

$$\begin{aligned}r_i^{\text{o}}(\omega_i)&=\frac{1}{\mathbb{E}[X_i]}\left(\mathbb{P}[Y_i+\omega_i<s_i]\right.\\&\qquad\cdot\mathbb{E}[X_i-Y_i-\omega_i|Y_i+\omega_i<s_i]\\&\qquad+\mathbb{P}[s_i<Y_i+\omega_i\le X_i]\\&\qquad\left.\cdot\mathbb{E}[X_i-Y_i-\omega_i|s_i<Y_i+\omega_i\le X_i]\right)\\&=\frac{1}{\mathbb{E}[X_i]}\left((1-\exp(-\lambda_i(s_i-\omega_i)))\right.\\&\qquad\cdot\left(\frac{s_i-\omega_i}{1-\exp(-\lambda_i(s_i-\omega_i))}-\frac{1}{\lambda_i}+\frac{1}{\beta_i}\right)\\&\qquad\left.+\exp(-\lambda_i(s_i-\omega_i))\frac{\lambda_i}{\lambda_i+\beta_i}\frac{1}{\beta_i}\right)\\&=\frac{1}{\mathbb{E}[X_i]}\left(s_i-\omega_i+\exp(-\lambda_i(s_i-\omega_i))\frac{\beta_i}{\lambda_i(\lambda_i+\beta_i)}\right.\\&\qquad\left.-\frac{1}{\lambda_i}+\frac{1}{\beta_i}\right).\end{aligned}$$

Case 2: $\tau_i\le s_i\le\omega_i$

In this case, the expected hit ratio and cache occupancy of the caching-only policy are all 0 similar to Case 1. For the overhearing-only policy, we have

$$\begin{aligned}h_i^{\text{o}}(\omega_i)&=\mathbb{P}[\text{The next request for } d_i \text{ is a hit under } \pi^{\text{o}}(\omega_i)]\\&=\mathbb{P}[s_i<Y_i+\omega_i\le X_i]\\&=\exp\left(-\lambda_i(s_i-\omega_i)\right)\cdot\frac{\lambda_i}{\lambda_i+\beta_i},\\r_i^{\text{o}}(\omega_i)&=\frac{1}{\mathbb{E}[X_i]}\mathbb{P}[s_i<Y_i+\omega_i\le X_i]\\&\qquad\cdot\mathbb{E}[X_i-Y_i-\omega_i|s_i<Y_i+\omega_i\le X_i]\\&=\frac{1}{\mathbb{E}[X_i]}\exp(-\lambda_i(s_i-\omega_i))\frac{\lambda_i}{\lambda_i+\beta_i}\frac{1}{\beta_i}.\end{aligned}$$

Case 3: $s_i\le\tau_i\le\omega_i$

In this case, $h_i^{\text{o}}(\omega_i)$ and $r_i^{\text{o}}(\omega_i)$ are exactly the same as those in Case 2. As for $\pi^{\text{c}}(\tau_i)$, we have

$$\begin{aligned}h_i^{\text{c}}(\tau_i)&=\mathbb{P}[\text{The next request for } d_i \text{ is a hit under } \pi^{\text{c}}(\tau_i)]\\&=\mathbb{P}[X_i\le\tau_i]\\&=1-\exp(-\beta_i(\tau_i-s_i)).\end{aligned}$$

The expected cache occupancy can be calculated as

$$\begin{aligned}r_i^{\text{c}}(\tau_i)&=\frac{1}{\mathbb{E}[X_i]}\left(\mathbb{P}[X_i\ge\tau_i]\cdot\tau_i+\mathbb{P}[X_i<\tau_i]\cdot\mathbb{E}[X_i|X_i<\tau_i]\right)\\&=\frac{1}{\mathbb{E}[X_i]}(\mathbb{P}[X_i\ge\tau_i]\cdot\tau_i-\mathbb{P}[X_i\ge\tau_i]\cdot\mathbb{E}[X_i|X_i\ge\tau_i]\\&\quad+\mathbb{P}[X_i\ge\tau_i]\cdot\mathbb{E}[X_i|X_i\ge\tau_i]\\&\quad+\mathbb{P}[X_i<\tau_i]\cdot\mathbb{E}[X_i|X_i<\tau_i])\\&=\frac{1}{\mathbb{E}[X_i]}(-\mathbb{P}[X_i\ge\tau_i]\cdot\mathbb{E}[X_i-\tau_i|X_i\ge\tau_i]+\mathbb{E}[X_i])\\&=\frac{1}{\mathbb{E}[X_i]}\frac{1}{\beta_i}(1-\exp(-\beta_i(\tau_i-s_i))).\end{aligned}$$

Then applying Lemma 4 completes the proof. □

APPENDIX B
PROOF OF LEMMA 1

The proof of Lemma 1 consists of two steps. In Step 1, we will show that for $\tau_i\le s_i$, there exists $\widetilde{\omega}_i$ such that $h_i^{\text{co}}(\tau_i,\omega_i)\le h_i^{\text{o}}(\widetilde{\omega}_i)$ and $r_i^{\text{co}}(\tau_i,\omega_i)=r_i^{\text{o}}(\widetilde{\omega}_i)$; for $\tau_i>s_i$, there exists $\widetilde{\tau}_i$ such that $h_i^{\text{co}}(\tau_i,\omega_i)\le h_i^{\text{c}}(\widetilde{\tau}_i)$ and $r_i^{\text{co}}(\tau_i,\omega_i)=r_i^{\text{c}}(\widetilde{\tau}_i)$. Next, in Step 2, we will show that the $h_i^{\text{rco}}(\cdot)$ function characterizes the upper boundary.

Step 1: For $\tau_i\le s_i$, it is easy to observe that $\widetilde{\omega}_i=\omega_i$ will satisfy the property. By applying Theorem 1, we can verify that $h_i^{\text{co}}(\tau_i,\omega_i)=h_i^{\text{co}}(0,\omega_i)=h_i^{\text{o}}(\widetilde{\omega}_i)$ and $r_i^{\text{co}}(\tau_i,\omega_i)=r_i^{\text{co}}(0,\omega_i)=r_i^{\text{o}}(\widetilde{\omega}_i)$.

For $\tau_i>s_i$, we may first solve $\widetilde{\tau}_i$ as the unique solution to the equation $r_i^{\text{co}}(\tau_i,\omega_i)=r_i^{\text{c}}(\widetilde{\tau}_i)$. Then, we will prove that $h_i^{\text{co}}(\tau_i,\omega_i)\le h_i^{\text{c}}(\widetilde{\tau}_i)$. By applying Lemma 4, it is equivalent to prove $h_i^{\text{c}}(\tau_i)+h_i^{\text{o}}(\omega_i)\le h_i^{\text{c}}(\widetilde{\tau}_i)$.

Since $r_i^{\text{co}}(\tau_i,\omega_i)=r_i^{\text{c}}(\tau_i)+r_i^{\text{o}}(\omega_i)=r_i^{\text{c}}(\widetilde{\tau}_i)$, we have

$$r_i^{\text{o}}(\omega_i)=r_i^{\text{c}}(\widetilde{\tau}_i)-r_i^{\text{c}}(\tau_i)=\left(h_i^{\text{c}}(\widetilde{\tau}_i)-h_i^{\text{c}}(\tau_i)\right)/(\beta_i s_i+1),$$

where the second equation holds due to the fact that $\widetilde{\tau}_i>\tau_i>s_i$ and the linear relationship between h_i^{c} and r_i^{c}, which is illustrated in Fig. 6 and can be easily proved using Theorem 1. Moreover, Theorem 1 also indicates that $r_i^{\text{o}}(\omega_i)\ge$

$h_i^{\rm o}(\omega_i)/(\beta_i s_i+1)$. Therefore, we have $h_i^{\rm c}(\widetilde{\tau}_i) - h_i^{\rm c}(\tau_i) \geq h_i^{\rm o}(\omega_i)$, which completes Step 1.

Step 2: First of all, we will show that for any randomized item policy $\pi^{\rm rco}(\mathbf{q}_i, \boldsymbol{\tau}_i, \boldsymbol{\omega}_i)$ with $|\mathbf{q}_i| = |\boldsymbol{\tau}_i| = |\boldsymbol{\omega}_i| = n \geq 2$, there exist $\widetilde{\mathbf{q}}_i$, $\widetilde{\boldsymbol{\tau}}_i$ and $\widetilde{\boldsymbol{\omega}}_i$ with $|\widetilde{\mathbf{q}}_i| = |\widetilde{\boldsymbol{\tau}}_i| = |\widetilde{\boldsymbol{\omega}}_i| = 2$, such that $h_i^{\rm rco}(\mathbf{q}_i, \boldsymbol{\tau}_i, \boldsymbol{\omega}_i) \leq h_i^{\rm rco}(\widetilde{\mathbf{q}}_i, \widetilde{\boldsymbol{\tau}}_i, \widetilde{\boldsymbol{\omega}}_i)$ and $r_i^{\rm rco}(\mathbf{q}_i, \boldsymbol{\tau}_i, \boldsymbol{\omega}_i) = r_i^{\rm rco}(\widetilde{\mathbf{q}}_i, \widetilde{\boldsymbol{\tau}}_i, \widetilde{\boldsymbol{\omega}}_i)$.

The item policy $\pi^{\rm rco}(\mathbf{q}_i, \boldsymbol{\tau}_i, \boldsymbol{\omega}_i)$ is a randomization of n deterministic item policies $\pi^{\rm co}(\tau_i^{(j)}, \omega_i^{(j)})$, $1 \leq j \leq n$. As shown in Fig. 10, we may plot the cache occupancies and the hit ratios achieved by the deterministic item policies $\pi^{\rm co}(\tau_i^{(j)}, \omega_i^{(j)})$, $1 \leq j \leq n$, in a two-dimensional Cartesian coordinate system where the x-axis represents the cache occupancy and the y-axis represents the hit ratio. Based on Theorem 2, the cache occupancy and hit ratio achieved by any randomized item policy $\pi^{\rm rco}(\mathbf{q}_i, \boldsymbol{\tau}_i, \boldsymbol{\omega}_i)$ must be in the convex hull of these n points (i.e., a convex polygon).

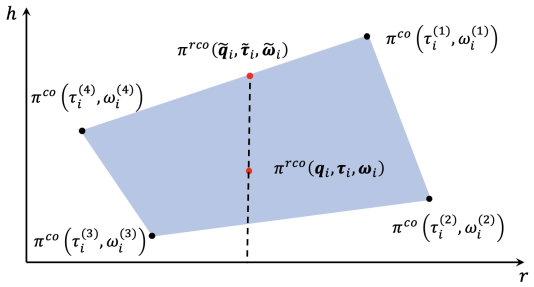


Fig. 10: Randomized policies achieving hit ratios and cache occupancies in a convex polygon.

Next, we may find $\pi^{\rm rco}(\widetilde{\mathbf{q}}_i, \widetilde{\boldsymbol{\tau}}_i, \widetilde{\boldsymbol{\omega}}_i)$ as the item policy that maximizes the expected overall hit ratio, while the cache occupancy is the same as the one achieved by $\pi^{\rm rco}(\mathbf{q}_i, \boldsymbol{\tau}_i, \boldsymbol{\omega}_i)$. We can observe that $\pi_i^{\rm rco}(\widetilde{\mathbf{q}}_i, \widetilde{\boldsymbol{\tau}}_i, \widetilde{\boldsymbol{\omega}}_i)$ must be on the boundary of the convex polygon and is achieved by the randomization of two deterministic policies. For example, in Fig. 10, $\pi_i^{\rm rco}(\widetilde{\mathbf{q}}_i, \widetilde{\boldsymbol{\tau}}_i, \widetilde{\boldsymbol{\omega}}_i)$ is a randomization of $\pi^{\rm co}(\tau_i^{(1)}, \omega_i^{(1)})$ and $\pi^{\rm co}(\tau_i^{(4)}, \omega_i^{(4)})$.

Therefore, for any item policy in the set $\Pi^{\rm rco}$, there must exist an item policy from the set

$$\widehat{\Pi}^{\rm rco} \triangleq \Big\{\pi^{\rm rco}\left(\left(q, 1-q\right), \left(\tau^{(1)}, \tau^{(2)}\right), \left(\omega^{(1)}, \omega^{(2)}\right)\right) : 0 \leq q \leq 1, 0 \leq \tau^{(j)} \leq \omega^{(j)}, 1 \leq j \leq 2\Big\}.$$

that achieves the same cache occupancy and a higher (or the same) hit ratio. An item policy in the set $\widehat{\Pi}^{\rm rco}$ is a randomization of two deterministic item policies $\pi^{\rm co}(\tau^{(1)}, \omega^{(1)})$ and $\pi^{\rm co}(\tau^{(2)}, \omega^{(2)})$. We have $\widehat{\Pi}^{\rm rco} \subset \Pi^{\rm rco}$. Based on the result of Step 1, we may know that the any point on the upper boundary of $\mathcal{R}_i^{\rm rco}$ must be achieved by a randomization of an overhearing-only item policy $\pi^{\rm o}(\omega_i)$ and a caching-only item policy $\pi^{\rm c}(\tau_i)$ for some $\omega_i \geq \tau_i \geq 0$. Based on the result of Step 1, we can further conclude that for any item policy in the set $\Pi^{\rm rco}$, there must exist an item policy from the set

$$\Big\{\pi^{\rm rco}\left(\left(q, 1-q\right), \left(\tau, 0\right), \left(+\infty, \omega\right)\right) : 0 \leq q \leq 1, 0 \leq \tau \leq +\infty, 0 \leq \omega \leq +\infty\Big\}$$

that achieves the same cache occupancy and a higher (or the same) hit ratio. By applying Theorem 1, we can show that $h_i^{\rm o}(r) \geq h_i^{\rm c}(r)$ for $0 \leq r \leq r_i^{\rm o}(0)$ and $h_i^{\rm o}(r_i^{\rm o}(0)) \leq h_i^{\rm c}(r_i^{\rm c}(+\infty)) = 1$. Therefore, for any item policy in the set $\Pi^{\rm rco}$, there must exist an item policy from the set

$$\widetilde{\Pi}^{\rm rco} = \{\pi^{\rm o}(\omega) : \omega \geq 0\} \cup \{\pi^{\rm rco}((q, 1-q), (+\infty, 0), (+\infty, 0)) : 0 \leq q \leq 1\}$$

that achieves the same cache occupancy and a higher (or the same) hit ratio.

Appendix C
Proof of Lemma 2

Consider the overhearing-only item policy $\pi^{\rm o}(\omega_i)$ with $\omega_i \geq s_i$ under the event-driven overhearing setting. Without loss of generality, we assume that the most recent request arrives at time 0. We will analyze the probability that the next request for d_i is a hit, and its expected cache occupancy. Let X_i denote the time when the next request for d_i arrives, and Y_i denote the time when we overhear d_i for the first time after the non-overhearing period (i.e., after time ω_i). We have

$$\begin{aligned} h_i^{\rm o}(\omega_i) &= \mathbb{P}[\text{The next request for } d_i \text{ is a hit under } \pi^{\rm o}(\omega_i)] \\ &= \mathbb{P}[Y_i + \omega_i \leq X_i] \end{aligned}$$

The cache occupancy of $\pi^{\rm o}(\omega_i)$ is

$$\begin{aligned} r_i^{\rm o}(\omega_i) =& \frac{1}{\mathbb{E}[X_i]}\mathbb{P}[Y_i + \omega_i \leq X_i] \\ & \cdot \mathbb{E}[X_i - Y_i - \omega_i | Y_i + \omega_i \leq X_i] \\ =& \frac{1}{\mathbb{E}[X_i]}\mathbb{P}[Y_i + \omega_i \leq X_i] \\ & \cdot \mathbb{E}[\mathbb{E}[X_i - Y_i - \omega_i | Y_i + \omega_i \leq X_i] | Y_i] \\ =& \frac{1}{\mathbb{E}[X_i]}\mathbb{P}[Y_i + \omega_i \leq X_i] \cdot \mathbb{E}\left[1/\beta_i | Y_i\right] \\ =& \mathbb{P}[Y_i + \omega_i \leq X_i]/(s_i\beta_i + 1). \end{aligned}$$

Therefore, we have $h_i^o(r) = (s_i\beta_i + 1)r$ for $\omega_i \geq s_i$, or equivalently, for $0 \leq r \leq r_i^{\rm co}(0, s_i)$.

Since the caching-only item policy $\pi^{\rm c}(\tau_i)$ is independent of the overhearing process, the hit ratio and the cache occupancy are the same as those in time-driving scenarios.

Appendix D
Proof of Lemma 3

We want to show that for a given cache size b, any achievable overall hit ratio must be no larger than $h^{\rm upper}$. Consider an idealized setting, where we can always overhear d_i and store it in the cache, $1 \leq i \leq N$, immediately after its OFF period. Let $h_i^{\rm ideal}(r)$ be the expected hit ratio of d_i when the cache occupancy is r under the overhearing only item policy $\pi^{\rm o}(\omega_i)$. $h_i^{\rm ideal}(r)$ is defined for $0 \leq r \leq 1/(\beta_i s_i + 1)$. We can show that $h_i^{\rm ideal}(r) = (\beta_i s_i + 1)r$. Note that a hit ratio 1 and a cache occupancy $1/(\beta_i s_i + 1)$ are achieved by $\pi^{\rm o}(s_i)$.

By applying Theorems 1 and 2, it is easy to prove that for any $(r, h) \in \mathcal{R}_i^{\rm rco}$, we have $h \leq h_i^{\rm ideal}(r)$. Therefore, the

maximal overall hit ratio of problem (3) should not exceed the maximal overall hit ratio of the following problem

$$\begin{aligned} \max_{r_i} \quad & \sum_{i=1}^{N} p_i \cdot h_i^{\text{ideal}}(r_i) \\ \text{subject to} \quad & 0 \le r_i \le 1, 1 \le i \le N, \\ & \sum_{i=1}^{N} r_i \le b. \end{aligned} \tag{17}$$

Based on Equation (2), we have $p_i \cdot h_i^{\text{ideal}}(r_i) = p_i(\beta_i s_i + 1)r_i = \beta_i r_i / (\sum_{j=1}^{N} 1/(s_j + 1/\beta_j))$. Since the data items are sorted such that β_i is decreasing with respect to i, the optimal solution to problem (17) is to set $r_i = 1/(s_i\beta_i+1)$ for $1 \le i \le K$, $r_{K+1} = b - \sum_{i=j}^{K} 1/(s_j\beta_j + 1)$ and $r_i = 0$ for $i > K+1$. And the achieved optimal overall hit ratio is

$$\sum_{i=1}^{K} p_i + p_{K+1}\left(\beta_{K+1}s_{K+1} + 1\right)\left(b - \sum_{i=1}^{K} \frac{1}{\beta_i s_i + 1}\right) \triangleq h^{\text{upper}},$$

which is an upper bound for the overall hit ratio that achieved by any feasible solution to problem (3). Therefore, we have $h^*(M) \le h^{\text{upper}}$ for $\forall M > 0$.

APPENDIX E
PROOF OF THEOREM 4

Consider M edge caches with event-driven overhearing opportunities. Assume that the data item d_i is served by the item policy $\pi^{\text{o}}(s_i)$. We define H_i as

$$H_i \triangleq \lim_{T\to\infty} \frac{\text{Number of hits for } d_i \text{ during } [0,T]}{\text{Number of requests for } d_i \text{ during } [0,T]}.$$

We first introduce the following lemma, where the proof the provided in the technical report [43].

Lemma 5. *Consider M edge caches with event-driven overhearing opportunities. If the data item d_i is served by the item policy $\pi^o(s_i)$, then we have $0 \le 1 - H_i \le 2\sqrt{(\beta_i s_i + 1)/M}$ almost surely.*

Proof of Theorem 4. First, define an overhearing only policy as follows. Let the overhearing only element policy $\pi^{\text{o}}(s_i)$ serve d_i, $1 \le i \le K$, where K in defined in (10). Based on the definition of K, we have $\sum_{i=1}^{K} r_i^{\text{o}}(s_i) \le b$, where $r_i^{\text{o}}(s_i)$ is the cache occupancy of d_i under $\pi^{\text{o}}(s_i)$. We serve d_{K+1} by $\pi^{\text{o}}(s_{K+1})$, if $\sum_{i=1}^{K+1} r_i^{\text{o}}(s_i) \le b$. Otherwise, we serve d_{K+1} by $\pi^{\text{o}}(\omega_{K+1})$, such that $r_{K+1}^{\text{o}}(\omega_{K+1}) = 1 - \sum_{i=1}^{K} r_i^{\text{o}}(s_i)$. We note that

- if d_{K+1} is served by $\pi^{\text{o}}(\omega_{K+1})$ with $\omega_{K+1} \neq s_{K+1}$ under the proposed overhearing only policy, then based on Lemma 2, we have $h_i^{\text{o}}(\omega_{K+1}) > h_i^{\text{upper}}$, where h_i^{upper} is the hit ratio of d_i under the idealized policy defined in Section IV-B;
- the proposed overhearing only policy cannot outperform $\boldsymbol{\pi}^E$, since the overhearing only policy is a feasible solution of problem (8), while $\boldsymbol{\pi}^E$ is the optimal solution.

Let $h_i^{\text{o}}(M) = \mathbb{E}[H_i]$ denote the expected hit ratio of d_i achieved by the proposed overhearing only policy. By applying Lemma 5 and the fact that H_i is bounded, we have

$$h_i^{\text{o}}(M) \ge 1 - 2\sqrt{(\beta_i s_i + 1)/M}.$$

Under the idealized policy proposed in Section IV-B, the hit ratios for d_i, $K+1 < i \le N$, must be zeros. Therefore, we can conclude that

$$\begin{aligned} h^{\text{upper}} - h^E(M) &\le h^{\text{upper}} - \sum_{i=1}^{N} p_i h_i^{\text{o}}(M) \\ &\le \sum_{i=1}^{K+1} p_i(1 - h_i^{\text{o}}(M)) \\ &\le \sum_{i=1}^{K+1} 2p_i\sqrt{(\beta_i s_i + 1)/M} \\ &\le \max_{1\le i\le K+1} 2\sqrt{(\beta_i s_i + 1)/M}. \end{aligned}$$

□

REFERENCES

[1] F. Qian, K. S. Quah, J. Huang, J. Erman, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "Web caching on smartphones: ideal vs. reality," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 2012, pp. 127–140.

[2] D. Niyato, D. I. Kim, P. Wang, and L. Song, "A novel caching mechanism for internet of things (iot) sensing service with energy harvesting," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

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

[4] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 80–87, 2018.

[5] Q. Jia, R. Xie, T. Huang, J. Liu, and Y. Liu, "Efficient caching resource allocation for network slicing in 5G core network," *IET Communications*, vol. 11, no. 18, pp. 2792–2799, 2017.

[6] S. Li, J. Xu, M. Van Der Schaar, and W. Li, "Popularity-driven content caching," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[7] 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.

[8] A. Cidon, A. Eisenman, M. Alizadeh, and S. Katti, "Dynacache: Dynamic cloud caching," in *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15)*, 2015.

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

[10] R. Nishtala, H. Fugal, S. Grimm, M. Kwiatkowski, H. Lee, H. C. Li, R. McElroy, M. Paleczny, D. Peek, P. Saab *et al.*, "Scaling Memcache at Facebook," in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 385–398.

[11] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, and M. Paleczny, "Workload analysis of a large-scale key-value store," in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, 2012, pp. 53–64.

[12] P. R. Jelenković, "Asymptotic approximation of the move-to-front search cost distribution and least-recently used caching fault probabilities," *Annals of Applied Probability*, pp. 430–464, 1999.

[13] G. Adomavicius and Y. Kwon, "Improving aggregate recommendation diversity using ranking-based techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 896–911, 2011.

[14] N. J. Hurley, "Personalised ranking with diversity," in *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013, pp. 379–382.

[15] X. Qian, D. Lu, Y. Wang, L. Zhu, Y. Y. Tang, and M. Wang, "Image re-ranking based on topic diversity," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3734–3747, 2017.

[16] J. Tadrous, A. Eryilmaz, and A. Sabharwal, "Action-based scheduling: Leveraging app interactivity for scheduler efficiency," *IEEE/ACM transactions on networking*, vol. 27, no. 1, pp. 112–125, 2018.

[17] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, "Wave: Popularity-based and collaborative in-network caching for content-oriented networks," in *2012 Proceedings IEEE INFOCOM Workshops*. IEEE, 2012, pp. 316–321.
[18] E. J. O'neil, P. E. O'neil, and G. Weikum, "The LRU-K page replacement algorithm for database disk buffering," *Acm Sigmod Record*, vol. 22, no. 2, pp. 297–306, 1993.
[19] E. Friedlander and V. Aggarwal, "Generalization of LRU cache replacement policy with applications to video streaming," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, vol. 4, no. 3, pp. 1–22, 2019.
[20] N. Beckmann, H. Chen, and A. Cidon, "LHD: Improving cache hit rate by maximizing hit density," in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, 2018, pp. 389–403.
[21] G. Quan, J. Tan, A. Eryilmaz, and N. Shroff, "A new flexible multi-flow LRU cache management paradigm for minimizing misses," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 2, pp. 1–30, 2019.
[22] G. Domingues, G. Mendonça, E. D. S. E. Silva, R. M. Leão, D. S. Menasché, O. Rottenstreich, M. Dehghan, and D. Towsley, "The role of hysteresis in caching systems," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, vol. 6, no. 1, pp. 1–38.
[23] A. Ferragut, I. Rodríguez, and F. Paganini, "Optimizing TTL caches under heavy-tailed demands," *ACM SIGMETRICS Performance Evaluation Review*, vol. 44, no. 1, pp. 101–112, 2016.
[24] M. Dehghan, L. Massoulie, D. Towsley, D. S. Menasche, and Y. C. Tay, "A utility optimization approach to network cache design," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1013–1027, 2019.
[25] J. Jung, A. W. Berger, and H. Balakrishnan, "Modeling TTL-based internet caches," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 1. IEEE, 2003, pp. 417–426.
[26] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, "Temporal locality in today's content caching: why it matters and how to model it," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 5, pp. 5–12, 2013.
[27] M. Garetto, E. Leonardi, and S. Traverso, "Efficient analysis of caching strategies under dynamic content popularity," in *2015 IEEE conference on computer communications (INFOCOM)*. IEEE, 2015, pp. 2263–2271.
[28] M. Leconte, G. Paschos, L. Gkatzikis, M. Draief, S. Vassilaras, and S. Chouvardas, "Placing dynamic content in caches with small population," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.
[29] D. T. Hoang, D. Niyato, D. N. Nguyen, E. Dutkiewicz, P. Wang, and Z. Han, "A dynamic edge caching framework for mobile 5G networks," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 95–103, 2018.
[30] K. Qi, S. Han, and C. Yang, "Learning a hybrid proactive and reactive caching policy in wireless edge under dynamic popularity," *IEEE Access*, vol. 7, pp. 120 788–120 801, 2019.
[31] S. Kumar and R. Tiwari, "Optimized content centric networking for future internet: dynamic popularity window based caching scheme," *Computer Networks*, vol. 179, p. 107434, 2020.
[32] J. Gao, S. Zhang, L. Zhao, and X. Shen, "The design of dynamic probabilistic caching with time-varying content popularity," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1672–1684, 2020.
[33] T. Zong, C. Li, Y. Lei, G. Li, H. Cao, and Y. Liu, "Cocktail edge caching: Ride dynamic trends of content popularity with ensemble learning," *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, pp. 208–219, 2022.
[34] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Single vs distributed edge caching for dynamic content," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 669–682, 2021.
[35] M. Bastopcu and S. Ulukus, "Information freshness in cache updating systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1861–1874, 2020.
[36] S. Zhang, L. Wang, H. Luo, X. Ma, and S. Zhou, "Aoi-delay tradeoff in mobile edge caching with freshness-aware content refreshing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 8, pp. 5329–5342, 2021.
[37] 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.
[38] 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.
[39] 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.
[40] 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.
[41] 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.
[42] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[43] G. Quan, A. Eryilmaz, and N. Shroff, "Optimal edge caching for individualized demand dynamics," *arXiv preprint arXiv:2310.14631*, 2023.

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 Awards. 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 holds or has held Visiting (chaired) Professor positions with Tsinghua University, Beijing, China, Shanghai Jiaotong University, Shanghai, China, and the Indian Institute of Technology Bombay, Mumbai, India. 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.