Fresh Caching of Dynamic Contents Using Restless Multi-Armed Bandits Over Wireless Access

Ankita Koley and Chandramani Singh, *Member, IEEE*

***Abstract*—We consider a dynamic content fetching, caching and delivery problem wherein the contents get updated at a central server, and local copies of a subset of the contents are cached at an edge cache associated with a base station (BS). Upon being requested, the BS can fetch the requested content from the central server and serve or can serve the locally cached version or can deny service. Fetching a content incurs a fixed fetching cost, serving a cached version incurs an ageing cost proportional to the age-of-version (AoV) of the content, and denying service incurs a fixed missing cost. Furthermore, due to unreliability of the channel between the BS and the users, a content delivery can fail, again resulting in a missing cost. We formulate an optimal content fetching, caching and delivery problem to minimize the average cost subject to the cache capacity constraint. This problem belongs to the class of continuous time restless multi-armed bandit (RMAB) problems with state dependent feasible action sets. We show that the single content problem is indexable, derive its Whittle indices and provide a Whittle index based policy for the original multi-content problem. Finally, we numerically evaluate the performance of the proposed policy and compare it to the existing works. We demonstrate that the proposed policy substantially outperforms the existing ones and, performance-wise, is very close to the optimal policy.**

***Index Terms*—Content Distribution Networks (CDNs), caching, age of information, restless multi-armed bandits (RMAB), Whittle indices.**

I. Introduction

OVER the past few years, online social networks and digital platforms like Facebook, Instagram, LinkedIn, and YouTube have become viral platforms for users to interact, communicate, and share contents. Their increasing popularity has resulted in a huge volume of content being shared across these platforms. Their Content Distribution Networks (CDNs) deploy caches in various geographical locations closer to the users in addition to the central server to ensure timely delivery of the contents; for example, the Facebook CDN uses several layers of caches along with the backend server [1]. Distributed caching reduces not only latency for the end users but also backhaul traffic and the resulting congestion, specifically during peak hours. Depending on contents' evolution on online platforms, their relevance changes, i.e., the earlier versions become less relevant, for example, consider Facebook's news feed or YouTube's recommendation page. The servers hold the most recent versions of the contents. As new versions of the contents appear at central servers, these also need to be updated at the edge caches [2].

Delivery of fresh versions of the contents is crucial in the above applications. The cached versions of the contents become less relevant on appearance of their updates at the central server. So, upon receiving a request for a cached content, the cache may serve the cached version or may fetch a fresh version and serve or may deny service, depending on the *freshness* of the cached version. Age of Information (AoI) is a widely adopted metric to measure information freshness [3], [4], [5]. However, it does not accurately capture the relevance of the contents. Contents with different AoI may have widely different relevance depending on their update statistics. Abolhassani et al. [6] have introduced a new metric called age-of-version (AoV) that captures the number of updates of a content at the server since the last cached version. The edge cache may want to fetch fresh versions of the contents and replace the cached ones depending on their AoVs. However, the edge caches are typically unaware of the updates occurring at the server. Specifically, they observe the AoVs of the cached contents only upon fetching the fresh versions.

We consider an edge cache attached to a base station (BS), connected to the server via the Internet, and serving a population of end users. Upon receiving a request for an uncached content, the cache may fetch the content from the server and serve or may deny service. In the former case, it may cache the fetched content by evicting another cached content. Upon receiving a request for a cached content, the cache may serve the cached version or may fetch a fresh version and serve or may deny service. Content delivery to the users may also fail owing to poor network conditions. There are costs associated with fetching the contents, serving stale versions and not serving either deliberately or due to poor network. We aim to design a content fetching, caching and delivery policy to minimize the average costs accounting for the limited cache capacity. This problem falls in the class of restless multi-armed bandit (RMAB) problems where each content acts as an arm. We propose a Whittle index based policy which performs very close to the optimal policy.

Dynamic content fetching and caching, in particular, the task of determining which contents to cache and when to replace the cached versions with fresh ones, is in general a challenging problem. It depends on several factors, e.g., limited cache capacity, dynamic content updates, request dynamics,

Received 5 December 2024; revised 25 July 2025; accepted 23 September 2025; approved by IEEE Transactions on Networking Editor R. Pedarsani. Date of publication 14 October 2025; date of current version 5 January 2026. This work was supported by Centre for Networked Intelligence at Indian Institute of Science, a CISCO CSR initiative. *(Corresponding author: Ankita Koley.)*

The authors are with the Department of ESE, Indian Institute of Science, Bangalore, Karnataka 560012, India (e-mail: ankitakoley@iisc.ac.in; chandra@iisc.ac.in).

This article has supplementary downloadable material available at https://doi.org/10.1109/TON.2025.3616225, provided by the authors.

Digital Object Identifier 10.1109/TON.2025.3616225

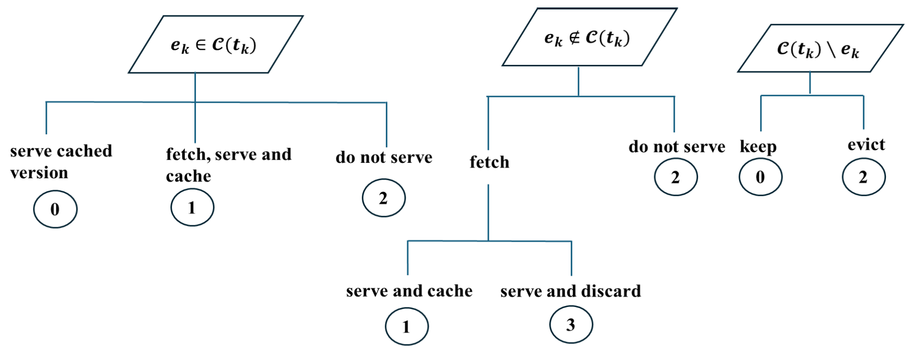


Fig. 1. Possible actions for different contents in $\mathcal{C}(t_k) \cup \{e_k\}$ when Content e_k is requested at time t. The actions are numbered from 0 to 3. For instance, Action 0 is applicable to the cached contents and has different annotations for the requested content and other contents in $\mathcal{C}(t_k)$. For the requested content it means *serve cached version* whereas for other contents it means *keep*. Recall, serving the requested content can be failed to to channel failure, resulting in a missing cost.

reliability of the wireless network between the edge cache and users, and the costs associated with content fetching, staleness and denying service.

A. Related Work

We review existing literature in three areas closely related to our work: AoV-based content fetching, caching over wireless networks and restless multi-armed bandits (RMAB).

1) AoV-Based Content Fetching and Caching: Yates [7] focuses on determining the average AoV at each node in a gossiping network. Delfani and Pappas [8] concentrate on minimizing the average AoV at all the nodes in such networks, assuming full knowledge of AoV at each node. In IoT systems, these approaches are especially meant to enhance data freshness. However, in edge caching systems, the edge cache is typically unaware of the updates at the central server, introducing an additional challenge in maintaining data freshness (minimizing AoV) at the edge cache.

The AoV of a content at the edge cache is learnt either by fetching the content [6] or employing a cache check [9]. In a single-cache framework, a cache check detects updates, incurring a cost. When new updates are found, the cached data is replaced with the latest version. In this model, upon being requested, the cache can only serve the cached version and fetches a new version only if the requested content is not cached. A more flexible framework allows fetching updated contents even when those are already cached to avoid serving stale versions [6], [10]. Abolhassani et al. [6] propose a static policy and show that it is asymptotically optimal. In [11], we propose a Whittle index based dynamic policy which outperforms the one proposed in [6]. These works do not consider unreliability of the wireless network between the edge cache and the end users.

2) Caching Strategies in Wireless Networks: The frameworks of [6] and [9] have been extended to a wireless edge [12] and to multiple caches connected to the server through a broadcast wireless edge [13], [14]. These works consider a wireless channel between the edge cache and the backend server, introducing possibility of content fetching failure due to poor channel conditions. Few works have considered unreliable wireless links between the edge cache and the end users. For example, Chen and Liew [15] study latency minimization considering caching of static contents over a wireless network. Zhang et al. [16] study average AoI-delay trade-off under AoI constraints with unlimited cache capacity. In contrast, we focus on optimal content fetching, caching and delivery of dynamic contents subject to limited cache capacity constraints.

3) Restless Multi-Armed Bandits: We frame the content fetching, caching and delivery problem subject to the cache capacity constraint as a RMAB where each content represents an arm. The Whittle index-based policies, are known to be asymptotically optimal for RMABs. However, establishing indexability and computing the Whittle indices are hard in general. Very few works have offered explicit characterizations of Whittle indices for their RMAB problems [17], [18], [19]. Several recent works have proposed efficient algorithms to numerically compute Whittle indices [20], [21]. More over, the existing works on RMABs (e.g., [15], [18], [19], [22], [23]) are primarily limited to problem with binary actions for each arm, pulling or not pulling it. In contrast, our problem entails more choices at each decision epoch (see Figure 1) which makes the analysis more challenging.

B. Our Contribution

We pose an optimal content fetching, caching and delivery problem to minimize the average content fetching, ageing and missing costs subject to the cache capacity constraint. To the best of our knowledge, this is the first work on optimal content caching and delivery that considers missing costs due to service denials and failures of the wireless channel between the cache and the users. We formulate the problem as a RMAB problem where each content is an arm and the single content problem is a partially observable Markov decision process (POMDP). Following are our main contributions.

1) We derive the optimal solution to the single content fetching, caching and delivery problem with a holding cost. We establish its indexability and obtain closed form expressions of the Whittle indices.
2) We propose a Whittle index based policy for the original multi-content problem. Our analysis also offers a lower bound on the optimal cost. We use it to demonstrate that the proposed policy performs very close to the optimal policy. We also show that it substantially outperforms the policies proposed in literature [6], [12].

3) Finally, we assume a reliable channel between the BS and the users and no service denial option. In this case, we compare performance of our proposed policy to that of the policy based on a distributed cache framework [13]. We demonstrate that our policy outperforms the one proposed in [13] also.

II. System Model

Let us consider a communication network with a central server, a Base Station (BS) associated with a edge cache and an end user population. The central server hosts N dynamic contents which are requested by the end users. The BS is connected to the central server via a wired network and the users are connected to the BS via wireless channels. The edge cache can fetch and store up to M contents from the server, and serve them locally to users upon request.

A. Content Dynamics

All the N contents are updated according to independent Poisson processes with λ_n being the update rate of the n^{th} content. The server always hosts the latest version of the contents.

B. Request Dynamics

The aggregate request process of the end users is a Poisson process with rate β. Each request could be for the n^{th} content with probability p_n independently of the other requests. Here, $p_n, 1 \leq n \leq N$ denote the relative popularity of the contents and $\sum_n p_n = 1$. For instance, the popularity of the contents on Web is widely modelled using *Zipf's distribution* wherein $p_n \propto 1/n^{\alpha}$ for the n^{th} most popular content [24]. Under the proposed request dynamics, n^{th} content's request rate is a Poisson process with rate $p_n\beta$. Let $\mathcal{C}(t) \subset \{1, \cdots, N\}$ denote the set of locally cached contents at time t. A subset of the cached contents could be updated at the central server once or multiple times since they were last fetched. Let $\nu_n(t) \geq 0$ be the number of times Content $n \in \mathcal{C}(t)$ has been updated since it was last fetched until time t.[1] We refer to $\nu_n(t)$ as the age-of-version (AoV) of Content n at time t. Note that the AoVs of the cached contents are not observable at the BS.

C. Wireless Channel Dynamics

Let $G(t) \in \{0, 1\}$ denote the state of the wireless channel between the BS and an arbitrary user at time t. If $G(t) = 1$ then the BS can successfully transmit the content to the user at time it and if $G(t) = 0$ then it cannot. We assume that the channels between the BS and different users are identically distributed. In particular, we let $G(t), t \geq 0$ be Bernoulli(q). The wireless channel states are also not observable at the BS.

D. Content Fetching, Ageing and Missing Costs

If a content, say Content n, is requested at time t, one of the following scenarios may occur.

(a) Content n is found at the edge cache. In this case, the edge cache can serve the cached version. However, the users detest receiving stale versions of the contents which is captured via *ageing costs*. We assume that serving a cached content incurs an ageing cost $c_a\nu_n(t)$ where c_a is the ageing cost per content update. Alternatively, the edge cache can also fetch the latest version of content n and serve, incurring a constant *fetching cost* c_f. The newly fetched copy replaces the existing one in the cache. Finally, the edge cache can also deny serving the request incurring a missing cost c_m. If the serving attempt fails due to the wireless channel state $G_k(t)$ being 0, then also a missing cost c_m is incurred. In this case, no ageing cost is incurred.

(b) Content n is not found at the edge cache. Then the edge cache can fetch the content at a cost c_f and serve. The fetched content either can replace an existing content in the cache or can be discarded after serving. Alternatively, the edge cache can deny serving the request incurring a cost c_m. If the edge cache cannot serve the content due $G_k(t)$ being 0, them also a cost c_m is incurred.

Remark 1: The content fetching cost can be measured in terms of the cost of procuring it from the server and the associated latency. The ageing cost reflects the negative impact of using stale data which leads to degradation of quality of experience (QoE) of the users. The QoE is often inferred from the user engagement metrics, such as click through rate (CTR) or watch time (WT) in platforms like YouTube, Facebook, Web Commerce, etc. In these scenarios, the ageing cost captures the reduction of the average CTR or WT. The missing cost is incurred when the cache denies service or is unable to serve the content due to downlink channel failure. It represents the additional latency and operational costs.

We assume same content fetching, ageing and missing costs for all the contents for notaional simplicity. Our analysis and the algorithm also apply when these parameters are different for different contents.

Let $t_k \geq 0$ and $e_k \in \{1, \cdots, N\}$ denote the k^{th} request epoch and identity of the requested content at t_k, respectively. Let G_k be the state of the wireless channel between the edge cache and the requesting user at t_k with $\mathbb{E}[G_k] = q$. Let ν_k^n be the AoV of content n at t_k. The edge cache does not have access to the AoV of the contents. However, the edge cache knows the rate (λ_n) at which the contents are getting updated. Let us represent various actions using numbers $0, 1, 2$ and 3 whose annotations are as in Figure 1. Further, let a_k^n denote the action at t_k vis a vis Content $n \in \mathcal{C}(t_k) \cup \{e_k\}$; $a_k^{e_k} \in \{0, 1, 2\}$ if $\{e_k\} \in \mathcal{C}(t_k)$ or the requested content is cached, $a_k^{e_k} \in \{1, 2, 3\}$ if $e_k \notin \mathcal{C}(t_k)$ or the requested content is not cached, and $a_k^n \in \{0, 2\}$ otherwise. We illustrate the annotations of actions in Figure 1.

Let C_k denotes the cost at t_k. Hence,

$$C_k = \mathbb{1}_{\{a_k^{e_k} \in \{1,3\}\}} c_f + G_k \mathbb{1}_{\{a_k^{e_k} = 0\}} c_a \nu_k^{e_k} + \left((1 - G_k) \mathbb{1}_{\{a_k^{e_k} \in \{0,1,3\}\}} + \mathbb{1}_{\{a_k^{e_k} = 2\}} \right) c_m \quad (1)$$

Further, let $A(T)$ denote the total number of requests until time T; $A(T) = \max\{k : t_k \leq T\}$ and $\mathbb{E}[A(T)] = \beta T$.

[1] We use the phrases "n^{th} content" and "Content n" interchangeably.

E. The Optimal Content Fetching, Caching and Delivery Problem

The optimal content fetching, caching and delivery problem taking actions that minimize the time average content fetching, ageing and missing costs subject to edge cache capacity constraints. It can be expressed more precisely as

$$\min \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{k=1}^{A(T)} C_k\right] \tag{2}$$

$$\text{s. t.} \sum_{n=1}^{N} \mathbb{1}_{\{a_k^n \in \{0,1\}\}} = M, \, \forall k \geq 1 \tag{3}$$

Remark 2: Treating $(\mathcal{C}(t_k), e_k, \nu_k^n, n \in \mathcal{C}(t_k)), (a_k^n, n \in \mathcal{C}(t_k) \cup \{e_k\})$ and $(e_{k+1}, \nu_{k+1}^n, n \in \mathcal{C}(t_{k+1}))$ as the state, the action and the random noise, respectively, at t_k, we can pose the above problem as a MDP.

Since the AoV of the contents or ν_k^n's are not observable at the edge cache before taking any actions. The problem (2) falls under the class of POMDP. This POMDP suffers from curse of dimensionality. However, we notice that the actions and state evolutions of different contents are only coupled through the cache capacity constraint (3). Hence, we formulate the problem as continuous time restless multi-armed bandit process (RMAB) and Whittle index policy [25] generally performs well as a heuristic policy to RMAB problems. Observe that, unlike the classical RMAB setting with binary actions, we have more state dependent actions (see Figure 1), rendering the problem complex.

1) Whittle Index Policy: We consider the following relaxed constraint instead of the hard constraint (3):

$$\lim_{T\to\infty} \frac{1}{\beta T}\mathbb{E}\left[\sum_{k=1}^{A(T)} \sum_{n=1}^{N} \mathbb{1}_{\{a_k^n \in \{0,1\}\}}\right] = M \tag{4}$$

Hence, we can write the Lagrangian of the problem (2) subject to the relaxed constraint (4) with multiplier C_h as follows:

$$\begin{aligned}
&\lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{k=1}^{A(T)} C_k\right] \\
&+C_h\left(\lim_{T\to\infty} \frac{1}{\beta T}\mathbb{E}\left[\sum_{k=1}^{A(T)} \sum_{n=1}^{N} \mathbb{1}_{\{a_k^n \in \{0,1\}\}}\right] - M\right) \\
&= \sum_{n=1}^{N} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{k=1}^{A(T)} C_k \mathbb{1}_{\{e_k = n\}} + \frac{C_h}{\beta}\mathbb{1}_{\{a_k^n \in \{0,1\}\}}\right] - C_h M \\
&=: \sum_{n=1}^{N} V^n(C_h) - C_h M.
\end{aligned} \tag{5}$$

Minimizing (2) subject to (4) entails first minimizing (5) for all C_h and then maximizing the optimal values, say $V(C_h)$, over all C_h. Let the optimal solution to the relaxed problem (5) be $\bar{V}$. The optimal policy for the relaxed problem does not offer a feasible policy for the original problem as it does satisfy the hard capacity constraints at each t_k. However, this policy provides "Whittle indices" associated with different states of all the contents which in turn can be used to design a feasible policy for the original problem. If all the contents are indexable the Whittle index policy computes the Whittle indices for each content and choose M contents to cache having highest indices. We formally define indexability of a content and Whittle index policy fin Section IV. We further denote $\hat{V}$ and V_W to be the optimal cost and the cost under Whittle index policy of the original problem, i.e, (2) subject to the hard constraint (3). Then, it is easy to show that $\bar{V} \leq \hat{V} \leq V_W$ [26]. According to Whittle's conjecture [25], as the number of contents N grows to infinity and the cache size M also increases proportionally with N, the value under Whittle index policy approaches to the value of the optimal policy of the relaxed problem, i.e., the optimal content fetching, caching and delivery problem (2) subject to (4). In the following section we obtain the optimal policy of the single content problem with holding cost C_h.

Observe that in minimizing (5) the optimization problems corresponding to different contents are decoupled. Moreover, the Lagrange multiplier C_h can be interpreted as the holding cost per unit time for the cached contents. So, in Section III, we focus on a single content problem with holding cost C_h. Note that the optimal policy will always keep the content for $C_h < 0$. Hence, as in [27], we are interested in $C_h \geq 0$. We design a Whittle index based policy in Section IV.

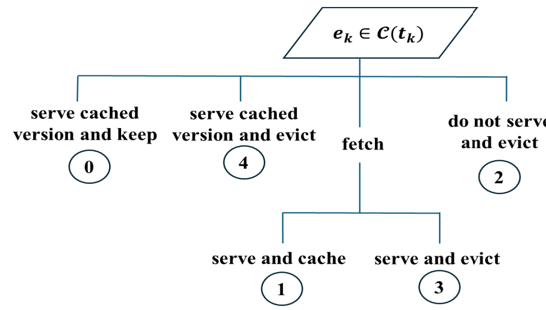


Fig. 2. Possible actions for the requested content, Content $e_k \in \mathcal{C}(t_k)$, in presence of the holding cost. The action *serve the cached version and evict* is added Figure 1. Annotations of the other actions have also changed.

III. Single Content Problem With Holding Cost

In this section, we focus on a single content problem with holding cost C_h. Suppose the content e_k is requested at t_k, and $e_k \in \mathcal{C}(t_k)$. Since there is a holding cost associated with keeping the content e_k in the cache, each action in $\{0, 1\}$ in Figure 1 can further be split into two actions specifying whether to keep or evict the content after serving. Therefore we introduce actions 3 and 4 as *fetch, serve and evict* and *serve the cached version and evict*, respectively, in Figure 2. Furthermore, it can be argued that for $C_h \geq 0$, if the optimal action is 2 or *do not serve* then it will be optimal to evict the content. Hence, we consider action 2 as *do not serve and evict* for the content $e_k \in \mathcal{C}(t_k)$. For $e_k \notin \mathcal{C}(t_k)$ and for $n \in \mathcal{C}(t_k) \setminus \{e_k\}$, the associated action sets remain as in Figure 1.

We now discuss the optimal content fetching and caching problem associated with a tagged content. We omit the content indices in the notation for brevity.

Let t_k be the k^{th} request epoch across all the contents. At t_k, the tagged content is requested with probability p. Recall that ν_k, G_k, and e_k are the AoV, channel state and requested content, respectively at t_k. We aim to minimize the following average cost for single content:

$$\lim_{T\to\infty}\frac{1}{T}\mathbb{E}\Bigg[\sum_{k=1}^{A(T)}\Big(G_k\mathbb{1}_{\{a_k\in\{0,4\}\}}c_a\nu_k+\mathbb{1}_{\{a_k\in\{1,3\}\}}c_f + \big((1-G_k)\mathbb{1}_{\{a_k\in\{0,1,3,4\}\}}+\mathbb{1}_{\{a_k=2\}}\big)c_m + \mathbb{1}_{\{a_k\in\{0,1\}\}}(t_{k+1}-t_k)C_h\Big)\Bigg] \tag{6}$$

The AoV of the content, ν_k can be observed only when the content is fetched from the central server. However, the average ageing cost at epoch k can be captured via the time elapsed since it was last fetched and cached. We denote this quantity as $\tau_k := t_k - \max\{t_l : l < k : a_l = 1\}$. The expected AoV at epoch k, $\mathbb{E}[\nu_k] = \lambda\tau_k$. Hence, we reformulate the above POMDP problem (6) as a Markov decision process considering τ_k instead of ν_k as a part of the state. Let B_k and Y_k be the two indicator variables for the tagged content. B_k indicates whether the requested content at t_k is the tagged content; $B_k = 1$ if e_k is the tagged content and 0 otherwise. If the tagged content is cached, then $Y_k = 1$ and 0, otherwise. We denote, $s_k = (\tau_k, Y_k, B_k)$ if $Y_k = 1$ and $s_k = (Y_k, B_k)$ if $Y_k = 0$. Hence, the state space is $\mathcal{S} := \{(\tau,1,B),(0,B),\tau\ge 0, B\in\{0,1\}\}$. The time interval between k^{th} and $(k+1)^{th}$ request epoch is exponentially distributed with parameter β.

Given $s_k = (\tau,1,B)$, at t_{k+1} the state will be as follows.

$$s_{k+1} = \begin{cases}(\tau+\Delta\tau,1,B'), & \text{if } a_k = 0\\ (\Delta\tau,1,B'), & \text{if } a_k = 1\\ (0,B'), & \text{if } a_k\in\{2,3,4\}\end{cases} \tag{7}$$

where $\Delta\tau \sim \exp(\beta)$ and $B' \sim \text{Bernoulli}(p)$. Given, $s_k = (0,1)$, at t_{k+1} the state will be as follows.

$$s_{k+1} = \begin{cases}(\Delta\tau,1,B'), & \text{if } a_k = 1\\ (0,B'), & \text{if } a_k\in\{2,3\}.\end{cases} \tag{8}$$

Recall from Figure 1 that action 0 is not applicable when the state is $(0,1)$ and no action is taken when the state is $(0,0)$. Given, $s_k = (0,0)$, s_{k+1} will be $(0,B')$. The expected single stage cost for a state action pair (s,a) is given by

$$c = (s,a) \begin{cases} qc_a\lambda\tau\mathbb{1}_{\{a\in\{0,4\}\}}+c_f\mathbb{1}_{\{a\in\{1,3\}\}}+\frac{C_h}{\beta}\mathbb{1}_{\{a\in\{0,1\}\}} \\ \big((1-q)\mathbb{1}_{\{a\in\{0,1,3,4\}\}}+\mathbb{1}_{\{a=2\}}\big)c_m, \text{if } s=(\tau,1,1)\\ c_f\mathbb{1}_{\{a\in\{1,3\}\}}+\frac{C_h}{\beta}\mathbb{1}_{\{a=1\}} \\ +\big((1-q)\mathbb{1}_{\{a\in\{1,3\}\}}+\mathbb{1}_{\{a=2\}}\big)c_m, \text{if } s=(0,1)\\ \frac{C_h}{\beta}\mathbb{1}_{\{a=0\}}, \text{if } s=(\tau,1,0)\\ 0, \text{if } s=(0,0).\end{cases} \tag{9}$$

Hence, the cost function under an admissible policy $\pi = \{\mu_0,\mu_1,\dots\}$ is

$$J_\pi(s)=\lim_{T\to\infty}\frac{1}{\mathbb{E}_\pi[A(T)]}\mathbb{E}_\pi\left[\sum_{k=1}^{A(T)}c(s_k,a_k)|s_0=s\right] \tag{10}$$

$$J_*(s)=\inf_{\pi\in\Pi}\lim_{T\to\infty}\frac{1}{\mathbb{E}_\pi[A(T)]}\mathbb{E}_\pi\left[\sum_{k=1}^{A(T)}c(s_k,a_k)|s_0=s\right] \tag{11}$$

where $a_k = \mu_k(s_k)$ and $J_*(s)$ is the solution under optimal policy. It can be shown from [28] that $J_\pi(s)$ and $J_*(s)$ are independent of s since the embedded discrete time Markov chains of the problems (10) and (11) have single recurrent classes. Moreover, Bellman's equation for the continuous time Markov decision problem is similar to the discrete-time problems [28, Chapter 5,Section 5.3]. Suppose, $h(s)$ and θ are the relative cost function of state s and optimal cost, respectively. Let us denote

$$L_r(\tau) = \int_r^\infty \beta e^{-\beta t}(ph(t+\tau,1,1)+(1-p)h(t+\tau,1,0))dt. \tag{12}$$

Then Bellman's equations from each state are as follows [28, Chapter 5,Section 5.3]:

$$h(\tau,1,1) = \min\Big\{qc_a\lambda\tau+(1-q)c_m+\frac{C_h-\theta}{\beta}+L_0(\tau), \; c_f+(1-q)c_m+\frac{C_h-\theta}{\beta}+L_0(0), \; qc_a\lambda\tau+(1-q)c_m-\frac{\theta}{\beta}+ph(0,1)+(1-p)h(0,0), \; c_f+(1-q)c_m-\frac{\theta}{\beta}+ph(0,1)+(1-p)h(0,0), \; c_m-\frac{\theta}{\beta}+ph(0,1)+(1-p)h(0,0)\Big\} \tag{13}$$

$$h(\tau,1,0)=\min\Big\{\frac{C_h-\theta}{\beta}+L_0(\tau), \; -\frac{\theta}{\beta}+ph(0,1)+(1-p)h(0,0)\Big\} \tag{14}$$

$$h(0,1)=\min\Big\{c_f+(1-q)c_m+\frac{C_h-\theta}{\beta}+L_0(0), \; c_f+(1-q)c_m-\frac{\theta}{\beta}+ph(0,1)+(1-p)h(0,0), \; c_m-\frac{\theta}{\beta}+ph(0,1)+(1-p)h(0,0)\Big\} \tag{15}$$

$$h(0,0) = -\frac{\theta}{\beta}+ph(0,1)+(1-p)h(0,0)$$
$$\Rightarrow h(0,0) = -\frac{\theta}{p\beta}+h(0,1). \tag{16}$$

The following lemma characterizes $h(\tau,1,1)$ and $h(\tau,1,0)$.

Lemma 1: (a) Both $h(\tau,1,1)$ and $h(\tau,1,0)$ are non-decreasing in τ.

(b) For a given $r\ge 0$, $L_r(\tau)$ is non-decreasing in τ.

Proof: (a) The proof of this lemma uses induction and relative value iteration and is standard [28].

TABLE I
STATES AND OPTIMAL ACTIONS IN CASE j ($j \in \{1, 2\}$)

Holding cost (C_h)	Optimal policy (π^*)			Optimal cost (θ)
	$s = (\tau, 1, 1)$	$s = (\tau, 1, 0)$	$s = (0, 1)$	
$C_h = 0$	$\pi^*(s) = \begin{cases} 0 \text{ for } \tau \leq \tau^* \\ 1 \text{ for } \tau > \tau^* \end{cases}$	$\pi^*(s) = 0$	$\pi^*(s) = 1$	$\theta = p\beta q c_a \lambda \tau^* + p\beta(1-q)c_m$
$0 < C_h \leq I_j$	$\pi^*(s) = \begin{cases} 0 \text{ for } \tau \leq \bar{\tau} \\ 4 \text{ for } \bar{\tau} < \tau \leq \tilde{\tau} \\ 1 \text{ for } \tau > \tilde{\tau} \end{cases}$	$\pi^*(s) = \begin{cases} 0 \text{ for } \tau \leq \bar{\tau} \\ 2 \text{ for } \tau > \bar{\tau} \end{cases}$	$\pi^*(s) = 1$	$\theta = p\beta q c_a \lambda \tilde{\tau} + p\beta(1-q)c_m$
$C_h > I_j$	Not applicable	Not applicable	$\pi^*(s) = \begin{cases} 3 \text{ if } j = 1 \\ 2 \text{ if } j = 2 \end{cases}$	$\theta = \begin{cases} p\beta(c_f + (1-q)c_m) \text{ if } j = 1 \\ p\beta c_m \text{ if } j = 2 \end{cases}$

(b) Since $h(\tau, 1, 1)$ and $h(\tau, 1, 0)$ are non-decreasing in τ, the integrand in $L_r(\tau)$ is also non-decreasing in τ. Hence $L_r(\tau)$ is non-decreasing in τ. ∎

We now state the optimal policy for the single content problem with holding cost.

Theorem 2: Define $\tau^0 := \frac{c_f}{qc_a\lambda}$, $\hat{\tau} := \frac{c_m}{c_a\lambda}$,

$$\tau^* := -\frac{1}{p\beta} + \sqrt{\left(\frac{1}{p\beta}\right)^2 + \frac{2c_f}{p\beta q c_a \lambda}}$$

Let $\bar{\tau}$ and $\tilde{\tau}$ are the solutions to the following two equations:

$$p\beta q c_a \lambda \left(\tilde{\tau}\bar{\tau} - \frac{\bar{\tau}^2}{2}\right) - C_h\bar{\tau} + qc_a\lambda\tilde{\tau} - c_f = 0 \quad (17)$$

$$\beta(\tilde{\tau} - \bar{\tau}) + e^{-\beta(\tilde{\tau} - \bar{\tau})} - 1 - \frac{C_h}{pqc_a\lambda} = 0. \quad (18)$$

Following are three cases and the optimal policy π^* in each of these.

1) $\frac{c_f}{q} \leq c_m$: Let us define

$$I_1 := p\beta c_f - pqc_a\lambda(1 - e^{-\beta\tau^0}).$$

Then the optimal actions are as described in Table I. Additionally, $0 \leq \bar{\tau} < \tau^* < \tilde{\tau} \leq \tau^0$.

2) $c_m < \frac{c_f}{q} \leq c_m + \frac{p\beta c_m^2}{2c_a\lambda}$. Let us define

$$I_2 := p\beta q c_a \lambda(\hat{\tau} - \bar{\tau}_{\min}) + pqc_a\lambda e^{\beta(\bar{\tau}_{\min} - \hat{\tau})} - pqc_a\lambda$$

where $\bar{\tau}_{min}$ is the solution to the following equation.

$$p\beta q c_a \lambda \frac{\bar{\tau}^2}{2} + pqc_a\lambda\bar{\tau}(1 - e^{\beta(\bar{\tau} - \hat{\tau})}) + qc_m - c_f = 0.$$

Again, the optimal actions are as in Table I. Additionally, $0 \leq \bar{\tau}_{\min} \leq \bar{\tau} < \tau^* < \tilde{\tau} \leq \hat{\tau}$.

3) $\frac{c_f}{q} > c_m + \frac{p\beta c_m^2}{2c_a\lambda}$. The optimally controlled Markov chain has only two recurrent states $(0, 1)$ and $(0, 0)$. The optimal action is 2 or *do not serve and evict* and the optimal cost $\theta = p\beta c_m$.

Remark 3: Theorem 2 establishes that the optimal policy for the single user problem is of threshold type for a given value of C_h where the threshold is on the time elapsed since a fresh version of the content is fetched and cached (τ). Moreover, it provides closed forms of the thresholds for different values of C_h in Table I. For sufficiently large c_m and $q = 1$, the optimal policy for the single content problem with holding cost (Theorem 2) becomes the same as [11, Theorem 2] where channel failures and the action *do not serve* have not been considered.

Proof: We outline the proof of this theorem. The detailed proof can be found in Appendix A (See the Supplementary Material). We consider the following two cases separately.

a. $\frac{C_h - \theta}{\beta} + L_0(0) \leq h(0, 0) + \min\{qc_m - c_f, 0\}$. In this case the optimally controlled Markov chain with state space $\mathcal{S}$ is ergodic. We show that this occurs if $0 \leq C_h \leq I_j$ for $j \in \{1, 2\}$ and if $0 \leq \frac{c_f}{q} \leq c_m + \frac{p\beta c_m^2}{2c_a\lambda}$.

b. $\frac{C_h - \theta}{\beta} + L_0(0) > h(0, 0) + \min\{qc_m - c_f, 0\}$. In this case the states $(\tau, 1, 0)$ and $(\tau, 1, 1)$ are transient whereas the states $\mathcal{S}' := \{(0, 0), (0, 1)\}$ constitute a recurrent communicating class. We show that this case occurs if either $\frac{c_f}{q} > c_m + \frac{p\beta c_m^2}{2c_a\lambda}$ or $C_h > I_j$ for $j \in \{1, 2\}$ and $0 \leq \frac{c_f}{q} \leq c_m + \frac{p\beta c_m^2}{2c_a\lambda}$.

We derive the optimal actions in each cases. In particular, in each of these cases we derive the ranges of C_h, the optimal policy and the optimal costs under three conditions, namely, $\frac{c_f}{q} \leq c_m$, $c_m < \frac{c_f}{q} \leq c_m + \frac{p\beta c_m^2}{2c_a\lambda}$ and $\frac{c_f}{q} > c_m + \frac{p\beta c_m^2}{2c_a\lambda}$ ∎

Remark 4 (Transient states): Note from Theorem 2 that the optimal action for the state $(0, 1)$ is *do not serve* if either $\frac{c_f}{q} > c_m + \frac{p\beta c_m^2}{2c_a\lambda}$ or $C_h > I_j$ for $j \in \{1, 2\}$ and $0 \leq \frac{c_f}{q} \leq c_m + \frac{p\beta c_m^2}{2c_a\lambda}$. Hence, given the initial state $(0, 1)$ or $(0, 0)$ the Markov chain under optimal policy never reaches $(\tau, 1, 0)$ or $(\tau, 1, 1)$. However, if the initial states are $(\tau, 1, 0)$ or $(\tau, 1, 1)$. Then the optimal action for $(\tau, 1, 0)$ will be action 2 or *evict* and the next state will be either $(0, 0)$ or $(0, 1)$. The optimal action for the state $(\tau, 1, 1)$ will be in $\{2, 3\}$ and the next state will be either $(0, 0)$ or $(0, 1)$. So, the states $(\tau, 1, 0))$ and $(\tau, 1, 1)$ are transient states.

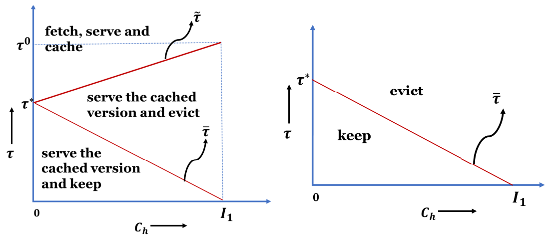
The following lemma characterizes $\bar{\tau}$ and $\tilde{\tau}$ in Theorem 2. We use this characterization in developing Whittle index based policy in Section IV.

Lemma 3:

1) The equations (17) and (18) are satisfied for unique non-negative values of $\tilde{\tau}$ and $\bar{\tau}$.
2) $\bar{\tau}$ and $\tilde{\tau}$ are decreasing and increasing function of C_h, for $0 < C_h \leq I_j$, where $j \in \{1, 2\}$ as in Theorem 2 respectively.

Proof: This proof is similar to the proof of [29, Lemma 3]. ∎

Remark 5 (Structure of the optimal policy): In Figures 3 and 4, we plot the threshold values of τ as we vary C_h and



(a) $s = (\tau, 1, 1)$ (b) $s = (\tau, 1, 0)$

Fig. 3. Optimal policy structure with respect to C_h for the state $(\tau, 1, 1)$ and $(\tau, 1, 0)$ when $\frac{c_f}{q} \leq c_m$.

indicate various regions depending upon optimal actions for the states $(\tau, 1, 1)$ and $(\tau, 1, 0)$, respectively. Figures 3 and 4 show the optimal policy structure for $\frac{c_f}{q} \leq c_m$ and $c_m < \frac{c_f}{q} \leq \frac{p\beta c_m^2}{2c_a\lambda}$, respectively.

(a) Figure 3a describes the optimal actions for the state $(\tau, 1, 1)$. When $C_h = 0$, it is optimal to keep the content $\forall\, \tau$. The optimal policy *serves the cached version* for $\tau \leq \tau^*$ and *fetches and serve*, otherwise. For $0 < C_h \leq I_1$, the optimal action is *serve the cached version and keep* for $\tau \leq \bar{\tau}$, *serve the cached version and evict* for $\bar{\tau} < \tau \leq \tilde{\tau}$, and *fetch, serve and cache* for $\tau > \tilde{\tau}$.

(b) Figure 3b describes the optimal policy of the state $(\tau, 1, 0)$. Note that, the optimal policy for the state $(\tau, 1, 0)$ can be derived from the optimal policy of the state $(\tau, 1, 1)$, in particular, from the line that represents $\bar{\tau}$ (see Figure 3a). For $C_h = 0$, it is always optimal to *keep*, for $0 < C_h \leq I_1$, the optimal action is *keep* for $\tau \leq \bar{\tau}$ and *evict* for $\tau > \bar{\tau}$. For $C_h > I_1$, the optimal action is *evict* for all τ.

In Figures 3a and 3b we observe that as C_h increases the set of τ for which the optimal action is *evict* expands and becomes $\mathbb{R}_+$ beyond I_1. We see similar trends in Figures 4a and 4b.

In the next section, we establish *indexability* of the single content problem with holding cost and provide explicit characterizations of the Whittle indices.

IV. Whittle Index Based Content Caching

In this section, we show that the single content MDP is indexable. We then compute the Whittle indices for each content $n \in \mathcal{C}(t_k) \cup e_k$. Finally, we provide a Whittle-index based policy for (2) subject to (3) in Section IV-A. We start with defining the notions of *passive sets* and *indexability*.

A. Passive Sets

The passive set of a content under the holding cost C_h is the subset of the states in $\mathcal{S}$ for which the optimal action is *evict, do not serve* or *discard* as applicable. Depending upon the relation between c_f and c_m we have the following as the passive sets:

1) If $\frac{c_f}{q} \leq c_m$, then $\mathcal{P}(C_h) := \{s : \pi^*(s) \in \{2, 3, 4\}\}$.
2) If $c_m < \frac{c_f}{q} \leq c_m + \frac{p\beta c_m^2}{2c_a\lambda}$, then $\mathcal{P}(C_h) := \{s : \pi^*(s) \in \{2, 4\}\}$.
3) If $\frac{c_f}{q} > c_m + \frac{p\beta c_m^2}{2c_a\lambda}$, then $\mathcal{P}(C_h) := \{s : \pi^*(s) = 2\}$.

B. Indexability

A content is called indexable if its passive set is nondecreasing in C_h [30], i.e., $\mathcal{P}(C_{h_1}) \subseteq \mathcal{P}(C_{h_2})$for $C_{h_1} \leq C_{h_2}$. The RMAB problem under consideration is called indexable if every content is indexable.

From Theorem 2, a content is never cached if the corresponding $\frac{c_f}{q} > c_m + \frac{p\beta c_m^2}{2c_a\lambda}$. Hence, in this case, the content's passive set is the whole state space for any value of C_h. In the following analysis, we assume that $\frac{c_f}{q} \leq c_m + \frac{p\beta c_m^2}{2c_a\lambda}$ for all the contents.

Theorem 4: Each content is indexable.

Proof: See Appendix B. (See the Supplementary Material) ■

From Theorem 4, the multi-content RMAB problem is also indexable. We now define the Whittle indices associated with a content.

1) Whittle Index: The Whittle index $W(s)$ associated with a state s of a content is the minimum holding cost that moves this state from the active set to the passive set. Equivalently, $W(s) = \min\{C_h : s \in \mathcal{P}(C_h)\}$.

In general the parameters $\bar{\tau}, \tau^*, \tilde{\tau}, \tau^0, \hat{\tau}$ etc. mentioned in Theorem 2 can be different for different contents. Hence the Whittle indices for different contents will also be different. Let $W^n(s)$ refer to the Whittle index associated with state s of content n. We now describe a scenario where the contents need to contend for the cache storage and how their Whittle indices could be used to resolve resolve this contention.

2) Whittle Index Based Caching: Let us consider the case where the requested content is not in the cache, i.e., $e_k \notin \mathcal{C}(t_k)$. If the edge cache fetches e_k to serve, it must also decide if e_k should replace a content in $\mathcal{C}(t_k)$ or e_k should be discarded after serving. In other words, the edge cache must decide which M of the $M+1$ contents $\mathcal{C}(t_k) \cup e_k$ should be kept in the cache. Towards this we compare the Whittle indices of the contents $C(t_k) \cup e_k$ for their respective states and keep the M contents with the highest indices in the caches. Each content $n \in \mathcal{C}(t_k)$ is in state $(\tau_k^n, 1, 0)$ and e_k is in the state $(0, 1)$. The following theorem provides explicit expressions for the Whittle indices $W^n(s)$ for the states $s \in \{(\tau^n, 1, 0), (0, 1)\}$.

Theorem 5: We consider the following three cases separately.

1) $\frac{c_f}{q} \leq c_m$: For $n \in \mathcal{C}(t_k)$

$$W^n((\tau^n, 1, 0)) = \begin{cases} C_h(\tau^n) & \text{if } \tau^n < \tau^{n*} \\ 0, & \text{if } \tau^n \geq \tau^{n*} \end{cases} \tag{19}$$

where $(C_h(\tau^n), \tilde{\tau}(\tau^n))$ is the unique solution to the equations

$$p_n\beta q c_a\lambda_n \left(\tilde{\tau}\tau^n - \frac{(\tau^n)^2}{2}\right) - C_h\tau^n + qc_a\lambda_n\tilde{\tau} - c_f = 0 \tag{20}$$

$$\beta(\tilde{\tau} - \tau^n) + e^{-\beta(\tilde{\tau} - \tau^n)} - 1 - \frac{C_h}{p_n c_a\lambda_n} = 0. \tag{21}$$

$$W^n((0, 1)) = p_n\beta c_f - p_n q c_a\lambda_n \left(1 - e^{-\frac{\beta c_f}{q c_a \lambda_n}}\right).$$

2) $c_m < \frac{c_f}{q} \leq c_m + \frac{p_n\beta c_m^2}{2c_a\lambda_n}$: For any n, $W^n(\tau^n, 1, 0)$ is same as in Case 3a for $\tau^n \geq \bar{\tau}_{min}$ and $W^n(\tau^n, 1, 0) = W^n(0, 1)$, otherwise, where

$$W^n(0,1) = p_n\beta q c_a\lambda_n(\hat{\tau}^n - \bar{\tau}_{\min}) + p_n q c_a\lambda_n e^{-\beta(\hat{\tau}^n - \bar{\tau}_{\min})} - p_n q c_a\lambda_n,$$

$\hat{\tau}^n = \frac{c_m}{c_a\lambda_n}$ and $\bar{\tau}_{min}$ is the unique solution to

$$p\beta q c_a\lambda_n\frac{\bar{\tau}^2}{2} + pqc_a\lambda_n\bar{\tau}(1 - e^{\beta(\bar{\tau} - \hat{\tau}^n)}) + qc_m - c_f = 0.$$

3) $\frac{c_f}{q} > c_m + \frac{p_n\beta c_m^2}{2c_a\lambda_n}$: For all n, $W^n(\tau^n, 1, 0) = W^n(0, 1) = 0$.

Proof: Let us consider a content $n \in \mathcal{C}(t_k)$ with state $(\tau^n, 1, 0)$. Suppose, $0 \leq \frac{c_f}{q} \leq c_m + \frac{p_n\beta c_m^2}{2c_a\lambda_n}$. Whittle index of content n is the minimum C_h for which $\pi^*(\tau^n, 1, 0) = 2$. From Theorem 2, $\pi^*(\tau^n, 1, 0) = 2$ if $\tau^n \geq \bar{\tau}(C_h)$ where $\bar{\tau}(C_h)$ is solution to (17) and (18). From Lemma 3, $\bar{\tau}(C_h)$ is decreasing in C_h. Hence, substituting $\bar{\tau} = \tau^n p = p_n$ and $\lambda = \lambda_n$ in (17) and (18) and solving for C_h gives minimum value of C_h for which $\pi^*(\tau^n, 1, 0) = 2$. In other words, Whittle index is the unique C_h that solves (20) and (21). The content $n = e_k$ has state $(0, 1)$. We note that from Table I, for $C_h \leq I_j^n$, the optimal action is 1 and for $C_h > I_j^n$, the optimal action is 3 for $j = 1$ and 2 for $j = 2$. Hence the Whittle index for e_k is I_j^n.

If $\frac{c_f}{q} > c_m + \frac{p_n\beta c_m^2}{2c_a\lambda_n}$, then from Theorem 2, it is always optimal to evict, and hence, $W^n(\tau^n, 1, 0) = W^n(0, 1) = 0$.■

C. Content Caching and Delivery Policy

We now describe the Whittle index based content fetching, caching and delivery policy. Towards this, let us revisit the set of available actions as shown in Figure 1.

(a) If the requested content is in the cache, i.e., $e_k \in \mathcal{C}(t_k)$, then e_k is in state $(\tau_k^{e_k}, 1, 1)$ whereas the other contents $n \in \mathcal{C}(t_k) \setminus e_k$ are in states $(\tau_k^n, 1, 0)$. In this case no content needs to be evicted irrespective of the action. Hence, the optimal actions are same as for of $C_h = 0$ in Theorem 2 (see Table I). If $\frac{c_f}{q} \leq \frac{p_{e_k}\beta c_m^2}{2c_a\lambda_{e_k}}$, then the optimal action for the requested content is
  (a) 0 or serve the cached copy if $\tau \leq \tau^{e_k*}$
  (b) 1 or fetch and cache if $\tau > \tau^{e_k*}$

  where $\tau^{e_k*} = -\frac{1}{p_{e_k}\beta} + \sqrt{\left(\frac{1}{p_{e_k}\beta}\right)^2 + \frac{2c_f}{p_{e_k}\beta q c_a\lambda_{e_k}}}$. The optimal actions for the other contents are 0. If $\frac{c_f}{q} > \frac{p_{e_k}\beta c_m^2}{2c_a\lambda_{e_k}}$, then the optimal action for the requested content is 2 or *do not serve* and for the other contents is 0.

(b) If the requested content is not in the cache, i.e., $e_k \notin \mathcal{C}(t_k)$, then each content $n \in \mathcal{C}(t_k)$ is in state $(\tau_k^n, 1, 0)$ whereas e_k is in state $(0, 1)$. The caching and eviction decisions are taken based on Whittle indices $W^n(\tau_k^n, 1, 0)$ for $n \in \mathcal{C}(t_k)$ and $W^{e_k}(0, 1)$ as described in Theorem 5. More precisely, the optimal actions for the contents $\mathcal{C}(t_k) \cup \{e_k\}$ are as follows.
  (a) If $\frac{c_f}{q} \leq c_m$ then fetch e_k and serve.
    i) If $W^{e_k}(0, 1) \geq \min_{n\in\mathcal{C}(t_k)} W^n(\tau_k^n, 1, 0)$, then cache e_k and evict a $n \in \arg\min_{n\in\mathcal{C}(t_k)} W^n(\tau_k^n, 1, 0)$.
    ii) If $W^{e_k}(0, 1) < \min_{n\in\mathcal{C}(t_k)} W^n(\tau_k^n, 1, 0)$ then discard e_k.
  (b) If $c_m < \frac{c_f}{q} \leq c_m + \frac{p_{e_k}\beta c_m^2}{2c_a\lambda_{e_k}}$
    i) If $W^{e_k}(0, 1) \geq \min_{n\in\mathcal{C}(t_k)} W^n(\tau_k^n, 1, 0)$ then fetch, serve and cache e_k and evict a $n \in \arg\min_{n\in\mathcal{C}(t_k)} W^n(\tau_k^n, 1, 0)$.
    ii) If $W^{e_k}(0, 1) < \min_{n\in\mathcal{C}(t_k)} W^n(\tau_k^n, 1, 0)$ then do not serve e_k.
  (c) If $\frac{c_f}{q} > c_m + \frac{p_{e_k}\beta c_m^2}{2c_a\lambda_{e_k}}$ then *do not serve* e_k.

1) Computation of Whittle Indices: To compute the Whittle indices (20) and (21) need to be solved for each $n \in \mathcal{C}(t_k)$. Recall that τ^n is the time elapsed since the content n is last fetched and cached. Observe from (20) that $\tilde{\tau}$ linear in C_h. We replace $\tilde{\tau}$ as a function of C_h in (21) to get an equation in C_h that has a unique solution. The solution to this equation is the Whittle index for the content n. For a given τ^n, the Whittle index can be computed with relative error $O(2^{-K})$ and complexity $O(log^3 K)$ using Newton's method [31]. The Whittle indices as a function of τ^n can be computed offline.

On the other hand we can tabulate discrete (approximate) values of the Whittle indices for discrete values of τ^n as follows. For each content n

1) Compute the range of potential values C_h, $[0, I_j^n]$, from Theorem 2.
2) For each discrete value of C_h in $[0, I_j^n]$ compute the corresponding values of τ^n and $\tilde{\tau}^n$, say $\tau^n(C_h)$ and $\tilde{\tau}^n(C_h)$, from (20) and (21).

Note that, (20) is a quadratic equation in τ^n for a given $\tilde{\tau}$, and so, it gives a closed form expression for τ^n in terms of $\tilde{\tau}$. On the other hand, (21) can be solved for $\tilde{\tau} - \tau^n$; it has computational complexity $O(log^3 K)$ where $O(2^{-K})$ is the allowed relative error [31]. The two solutions together give $\tilde{\tau}$ and τ^n. Hence, if $[0, I_j^n]$ is discretized into L_j^n values, the complexity of computing $\tau^n(C_h)$ at all the discrete points is $O(L_j^n \log^3 K)$. At any decision epoch, to obtain the Whittle index $W^n(\tau^n, 1, 0)$, we identify the closest $\tau^n(C_h)$ to the given τ^n (age of the content n), and then use the associated C_h as the Whittle index. The computation of Whittle indices is done offline. At each decision epoch, the policy merely compares the $M + 1$ indices of the cached and the requested contents. Hence, its complexity at each epoch is $O(M)$.

V. Single Vs. Distributed Cache

In this section we compare the performance of the proposed content fetching, caching and delivery policy to that of a multi-cache framework based policy in [13]. The authors in [13] propose a multi-cache and static caching framework to facilitate regular content fetching and delivery and also propose a policy based on this framework. We first describe their framework and their policy.

There is a set of K caches connected to a server via a wireless broadcast channel. When a cache fetches the content from the server, every cache receives the content with probability α, independently of the other caches. However, the caches have reliable connections to the users and cannot deny service upon being requested. These correspond to $q = 1$ and $c_m = \infty$ in our model in Section II. The content update and request arrival processes and the fetching and the ageing costs are as in our model. The proposed static caching framework is as follows. Each cache $k \in \{1, 2, \cdots, K\}$ stores a subset of the contents. Let $\mathcal{I}_n$ be the set of caches where content n is stored. Content requests are probabilistically split among the caches. More precisely, each request is directed to cache k with probability s_k where $\sum_{k=1}^{K} s_k = 1$. If a content n is requested, there can be two cases.

(a) The request is directed to the a cache $k \in \mathcal{I}_n$: In this case, cache k serves the cached version to the user, incurring an ageing cost $c_a \nu_n^k$ where ν_n^k is the AoV of content n at cache k.
(b) The request is directed to the a cache $k \notin \mathcal{I}_n$: In this case cache k fetches a fresh version of the content from the server. If the fetching attempt fails, cache k reattempts to fetch and continues to do so until a successful fetch happens, incurring an average fetching cost $\frac{c_f}{\alpha}$. Cache k discards content n after serving it. Other caches in $\{1, 2, \ldots, K\} \setminus \mathcal{I}_n$ also ignore it. But the caches in $\mathcal{I}_n$ update content n when they overhear its updated version.

Let us analyze the cache update rate in the above framework. Recall that a content n is fetched upon being requested if the request is directed to a cache in $\{1, 2, \ldots, K\} \setminus \mathcal{I}_n$, i.e., with probability $\left(1 - \sum_{j \in \mathcal{I}_n} s_j\right)$. Any cache in $\mathcal{I}_n$ updates content n on overhearing its fresh version. Hence the content update rate at any cache $k \in \mathcal{I}_n$ is $\mu_n = \beta p_n \left(1 - \sum_{j \in \mathcal{I}_n} s_j\right) p_u$ where p_u is the probability of content n being updated at cache k while being fetched by a cache $j \notin \mathcal{I}_n$. The probability of a successful fetch of content n at cache j requiring l attempts is $\alpha(1-\alpha)^{(l-1)}$ and the probability of content n being updated a cache k after l attempts by cache j is $(1-(1-\alpha)^l)$. Hence

$$p_u = \sum_{l=1}^{\infty} \alpha(1-\alpha)^{(l-1)}(1-(1-\alpha)^l) = \frac{1}{2-\alpha}.$$

and so,

$$\mu_n = \frac{\beta p_n}{2-\alpha} \left(1 - \sum_{k \in \mathcal{I}_n} s_k\right).$$

We would like to point out that the expression in [13, Equation (2)] misses the factor $(2 - \alpha)$. Note that there are no a priori restrictions on the number of caches or the cache sizes.

We now present the optimal number of caches, the optimal caching policy and the optimal cost under the above multi-cache framework, as analyzed in [13]. Following [13, Theorem 1], we define $y_n^* := \beta p_n \max\left\{0, 1 - \sqrt{\frac{\alpha(2-\alpha)c_a\lambda_n}{c_f \beta p_n}}\right\}$ for each content n. Without any loss of generality, assume that the contents are ordered such that $y_n^* \geq y_{n+1}^*$ for all $n \geq 1$. The optimal number of caches is

$$K^* = |\{n : y_n^* > 0\}| + 1. \tag{22}$$

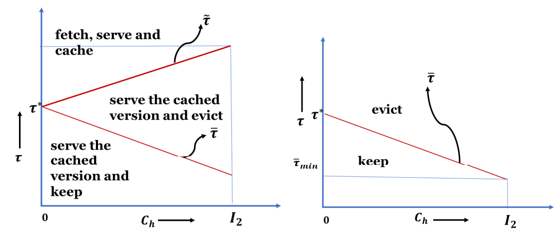
Content 1 is stored in caches $\{1, \cdots, K^*\}$, content 2 is stored in caches $\{2, \cdots, K^*\}$ and so on whereas no content is stored in cache K^*. More precisely, $\mathcal{I}_n = \{n, \cdots, K^* - 1\}$ for $1 \leq n \leq K^* - 1$ and $\mathcal{I}_n = \emptyset$ for all $n \geq K^*$. The optimal cost is

$$C^* = \frac{\beta c_f}{\alpha} - \sum_{n=1}^{K^*-1} \left(\sqrt{c_a \lambda_n (2-\alpha)} - \sqrt{\frac{c_f \beta p_n}{\alpha}}\right)^2.$$

Finally, an upper bound on the cache occupancy is

$$B^* = \frac{1}{2} K^* (K^* - 1).$$

We compare the cost of our Whittle index based policy for a single cache as proposed in Section IV-A to the cost of the above policy for the multi-cache framework. We keep the content update rates, the request arrival rates, and the ageing and fetching costs same. More precisely, we set $q = 1, c_m = \infty$ and the fetching cost to be $\frac{c_f}{\alpha}$ in our model. Moreover, we consider the cache capacity to be $K^* - 1$. Note that the caching policy in [13] caches $K^* - 1$ contents across $K^* - 1$ caches but requires a higher aggregate cache capacity B^*. In their policy, two different caches can have same or different versions of the same content. We demonstrate in Section VI-D that our Whittle index based policy outperforms the static policy in [13] (see Figure 13). It is worth noting that the policy in [13] might end up caching all the contents across as many caches as their framework does not put a restriction on cache capacity. For instance, if there are 1000 contents with $\lambda = 0.05$, $\beta = 40, c_f = 1, c_a = 0.1$, $\alpha = 0.9$ and $p_n \propto \frac{1}{n}$ then $K^* - 1 = 1000$. So, under the policy in [13], all the 1000 contents are cached across 1000 caches.



(a) $s = (\tau, 1, 1)$ (b) $s = (\tau, 1, 0)$

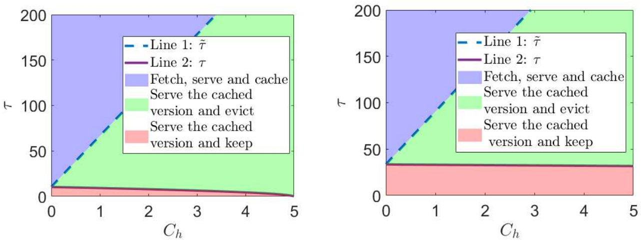
Fig. 4. Optimal policy structure with respect to C_h for the state $(\tau, 1, 1)$ and $(\tau, 1, 0)$ when $c_m < \frac{c_f}{q} \leq c_m + \frac{\beta c_m^2}{2 c_a \lambda}$.

VI. Numerical Results

In this section, we validate our analytical results via simulation. In particular, we plot the optimal policy structure of the single content problem for various parameters (Theorem 2). We the illustrate the performance of our Whittle index based policy by comparing it to a few existing policies. Finally, we explore how variations in the key parameters impact the average cost under the Whittle index based policy.

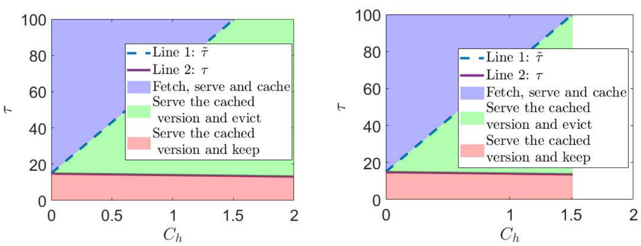
A. Optimal Policy Structure

We plot the optimal policy structure for a tagged content in states $(\tau, 1, 1)$ in Figure 5. Recall from Remark 5 that the optimal action for a state $(\tau, 1, 0)$ can be derived from the optimal action for the state $(\tau, 1, 1)$. The optimal action for the state $(0, 1)$ is straightforward from Theorem 2. We consider



(a) $c_f = 1, c_m = 2$ (b) $c_f = 10, c_m = 20$

Fig. 5. Optimal policy structure when $\frac{c_f}{q} \leq c_m$. The optimal action in the region between line 2 and C_h-axis, between line 1 and 2, and between line 1 and τ-axis are *serve the cached version and keep*, *serve the cached version and evict*, and *fetch, serve and cache*, respectively.



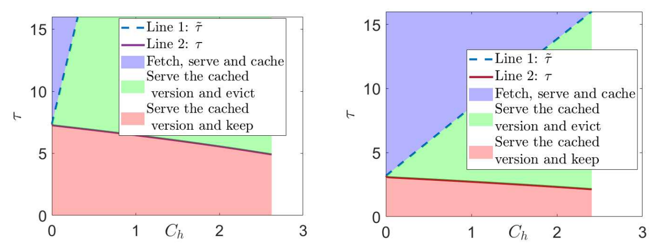
(a) $c_f = 2, c_m = 1$ (b) $c_f = 2, c_m = 0.5$

Fig. 6. Optimal policy structure when $c_m < \frac{c_f}{q} \leq c_m + \frac{p\beta c_m^2}{2c_a\lambda}$. The optimal action in the region between line 2 and C_h-axis, between line 1 and 2, and between line 1 and τ-axis are *serve the cached version and keep*, *serve the cached version and evict*, and *fetch, serve and cache*, respectively.

the request rate $\beta = 10$, the probability of requesting the tagged content $p = 0.5$, the ageing cost $c_a = 0.5$, the update rate of the content $\lambda = 0.01$ wireless channel state distribution parameter $q = 0.7$. We consider the following two cases.

1) $\frac{c_f}{q} \leq c_m$*:* This is case 1 in Theorem 2. We consider two sets of the parameters c_f and c_m, $(a)c_f = 1, c_m = 2$ and $(b)c_f = 10, c_m = 20$, both satisfying this case. From Theorem 2, τ^*, $\bar{\tau}$, $\tilde{\tau}$ and I_1 do not depend on c_m. So, changing c_m alone does not change the optimal policy as long as the condition $\frac{c_f}{q} \leq c_m$ is satisfied. We observe in Figures 5a and 5b that for a larger c_f the optimal policy is more inclined towards serving the cached content than fetching and serving the fresh version. In particular, the set of (C_h, τ) for which the optimal action action is *fetch, serve and cache* shrinks as c_f increases. From Remark 4, it is optimal to fetch, serve and discard the content for $C_h > I_1$. The values of I_1 corresponding to the settings (a) and (b) are 5 and 50, respectively. This implies that for a larger fetching cost the optimal policy is more inclined towards keeping the content in the cache than serving and discarding it.

2) $c_m < \frac{c_f}{q} \leq c_m + \frac{p\beta c_m^2}{2\ c_a\lambda}$*:* We fix $c_f = 2$ and consider two different values of c_m, 1 and 0.5. The values of I_2 corresponding to $c_m = 0.5$ and $c_m = 1$ are 1.5 and 3.28, respectively. So, the optimal action for $C_h \geq 1.5$ is to serve the content in Figure 6a and not to serve the content in Figure 6b. From Theorem 2, τ^*, $\bar{\tau}$ and $\tilde{\tau}$ are not affected by c_m if $C_h \leq I_2$. Hence, for $C_h \leq I_2$, the optimal policy does not depend upon c_m. Consequently, the regions corresponding to



(a) $c_a = 1$ (b) $c_a = 5$

Fig. 7. Optimal policy structure for different values of c_a. The optimal action in the region between line 2 and C_h-axis, between line 1 and 2, and between line 1 and τ-axis are *serve the cached version and keep*, *serve the cached version and evict*, and *fetch, serve and cache*, respectively.

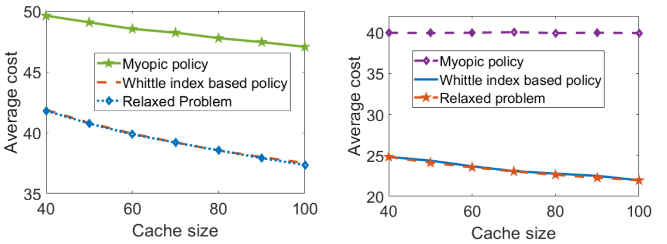
serve the cached version and keep, *serve the cached version and evict*, and *fetch, serve and cache*, are same in Figures 6a and 6b for $C_h \leq 1.5$. From Remark 4, for $C_h > I_2$, the optimal action is *do not serve* the content. However, I_2 is increasing in c_m. Hence, for a lower value of c_m, the optimal policy is more inclined towards not serving the content.

3) Optimal Policy Structure Under Different per Unit Ageing Costs c_a*:* We set $c_f = 1$, $c_m = 0.8, \lambda = 0.01$ and consider two different values of c_a, 1 and 5. Observe that these values satisfy $c_m < \frac{c_f}{q} \leq c_m + \frac{p\beta c_m^2}{2c_a\lambda}$. The average ageing cost increases with c_a, and so, the optimal policy is more inclined towards fetching the content and serving than serving the cached version. The set of (C_h, τ) for which the optimal action is *fetch, serve and cache* is smaller in Figure 7a than in Figure 7b. From Remark 4, the optimal action is *do not serve and evict* for $C_h \geq I_2$. Furthermore, I_2 decreases with c_a; $I_2 = 2.62$ for $c_a = 1$ and $I_2 = 2.4$ for $c_a = 5$. Hence, for a larger c_a the optimal policy is more inclined towards not serving and evicting the content.

B. Performance Evaluation

We consider the number of contents $N = 1000$, the update rate $\lambda_n = \lambda = 0.01$ for all n, the request rate $\beta = 40$ and the popularity of the contents being distributed as *Zipf* (1), i.e., $p_n \propto \frac{1}{n}$. We fix the fetching cost $c_f = 1$, per unit ageing cost $c_a = 0.01$, and vary the missing cost. The probability of wireless channel success $q = 0.7$. We use these parameters for performance analysis unless otherwise specified.

1) Optimality: We first compare the average cost under of problem (2) under the proposed Whittle index based policy to the optimal cost of the relaxed problem (5). For this we consider two cases (a) $\frac{c_f}{q} \leq c_m$ and (b) $\frac{c_f}{q} > c_m$ as the computation of Whittle indices are different for these cases (Theorem 5). We use $c_m = 2$ and $c_m = 1$ for cases (a) and (b), respectively. We numerically compute the average cost of the relaxed RMAB problem using Table I. We observe in Figures 8a and 8b that the average cost of problem (2) under the Whittle index based policy is very close to the optimal cost of the relaxed RMAB problem which acts as a lower bound of the optimal cost of problem (2). This suggests that the average cost under the Whittle index based policy is very close to the optimal cost. i.e., the Whittle index based policy is nearly optimal.



(a) $\frac{c_f}{q} \le c_m$ (b) $\frac{c_f}{q} > c_m$

Fig. 8. Performance of the Whittle index based policy. For each cache size, its cost is close to the optimal value of the relaxed RMAB problem and is much smaller than the cost of the myopic policy.

2) *Comparison to a Myopic Policy:* We compare the performance of the Whittle index based policy and a *myopic policy* based on [32, Section 4.3]. We first describe the *myopic policy* which aims to minimize the sum of the current stage costs and the terminal cost, treating the next state as the terminal state.

Let e_k be the content requested at t_k. One of the following two cases can happen.

a) $e_k \in \mathcal{C}(t_k)$*:* Then the myopic policy prescribes the action $a_k^* \in \arg\min_j C_{j,k}$ where $C_{j,k}$s are the costs corresponding to the actions $j \in \{0,1,2\}$ (see Figure 1). It can be seen that

$$C_{0,k} = qc_a\lambda\tau^{e_k} + (1-q)c_m + p_{e_k}\min\Big\{c_f, qc_a\lambda\Big(\tau^{e_k} + \frac{1}{\beta}\Big), qc_m\Big\} + p_{e_k}(1-q)c_m,$$

$$C_{1,k} = c_f + (1-q)c_m + p_{e_k}\min\left\{c_f, \frac{qc_a\lambda}{\beta}, qc_m\right\} + p_{e_k}(1-q)c_m,$$

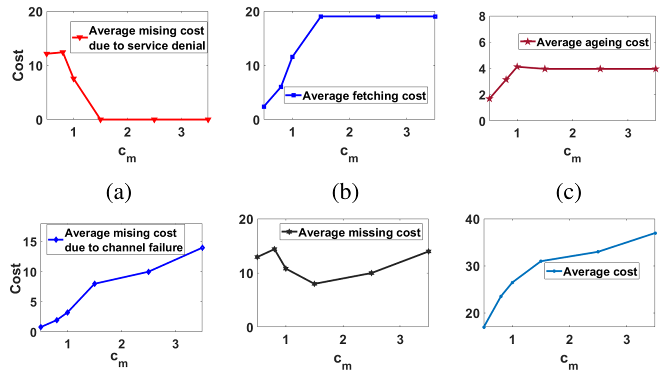
$$C_{2,k} = c_m + p_{e_k}\min\left\{c_f, qc_a\lambda\left(\tau^{e_k} + \frac{1}{\beta}\right), qc_m\right\} + p_{e_k}(1-q)c_m.$$

b) $e_k \notin \mathcal{C}(t_k)$*:* Then the myopic policy prescribes the action $a_k^* \in \arg\min_j C'_{j,k}$ where $C'_{j,k}$s are the costs corresponding to the actions $j \in \{1,2,3\}$ (see Figure 1). It can be seen that

$$C'_{1,k} = c_f + (1-q)c_m + \min_{n\in\mathcal{C}(t_k)}\Bigg\{p_n\min\{c_f, qc_m\} + \sum_{l\in\mathcal{C}(t_k)\setminus n} p_l\min\left\{c_f, qc_a\lambda\left(\tau^l + \frac{1}{\beta}\right), qc_m\right\} + \sum_{n\in\mathcal{C}(t_k)} p_n(1-q)c_m\Bigg\} + p_{e_k}\min\left\{c_f, \frac{qc_a\lambda}{\beta}, qc_m\right\} + p_{e_k}(1-q)c_m,$$

$$C'_{2,k} = c_m + \sum_{n\in\mathcal{C}(t_k)} p_n\left(\min\left\{c_f, qc_a\lambda\left(\tau^n + \frac{1}{\beta}\right), qc_m\right\} + (1-q)c_m\right) + p_{e_k}\min\{c_f, qc_m\} + p_{e_k}(1-q)c_m,$$

$$C'_{3,k} = c_f + (1-q)c_m + \sum_{n\in\mathcal{C}(t_k)} p_n\left(\min\left\{c_f, qc_a\lambda\left(\tau^n + \frac{1}{\beta}\right), qc_m\right\} + (1-q)c_m\right) + p_{e_k}\left(\min\{c_f, qc_m\} + (1-q)c_m\right).$$



(a) (b) (c) (d) (e) (f)

Fig. 9. Variation in the average cost and its components with respect to c_m.

If action 1 is taken for e_k then one of the contents in $\mathcal{C}(t_k)$ needs to be evicted. The myopic policy prescribes to evict a content n' where

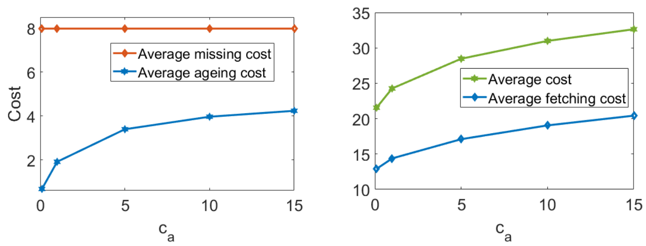
$$n' \in \arg\min_{n\in\mathcal{C}(t_k)}\Bigg\{p_n\min\{c_f, qc_m\} + \sum_{l\in\mathcal{C}(t_k)\setminus n} p_l\min\left\{c_f, qc_a\lambda\left(\tau^l + \frac{1}{\beta}\right), qc_m\right\}\Bigg\}.$$

We consider the same parameters as in Section VI-B1. We observe in Figures 8a and 8b that the Whittle index based policy performs much better than the myopic policy.

3) *Effect of the Missing Cost* c_m *on the Average Cost:* We set $q = 0.9, c_a = 10$, the cache size $M = 100$, and keep other parameters same as in Section VI-B. We consider $c_m = 0.5, 0.8, 1.0, 1.5, 2.5$ and 3.5. We observe the variations in different components of the average cost as we vary c_m. Varying c_m leads to one of the two cases, $\frac{c_f}{q} > c_m$ if $c_m < 1.1$ and $\frac{c_f}{q} \le c_m$ otherwise. We discuss these cases below.

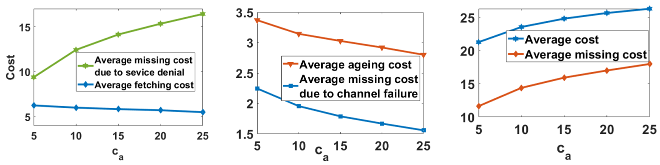
a) $\frac{c_f}{q} > c_m$*:* In this case, $\frac{c_f}{q}$ may also be larger than $c_m + \frac{p_n\beta c_m^2}{2c_a\lambda_n}$ for a few contents. The Whittle index based policy *does not serve* these contents (see Section IV-A). Hence, these contents incur only the missing costs, not the ageing and fetching costs. The number of contents for which $\frac{c_f}{q} > c_m + \frac{p_n\beta c_m^2}{2c_a\lambda_n}$ reduces as c_m increases, resulting in an increase in the number of services provided either by serving cached contents or by fetching and serving fresh versions. Consequentially, as c_m increases the average missing cost due to denying the requests (*do not serve* action) decreases (Figure 9a), the average fetching and ageing costs increase (Figures 9b and 9c), and the average missing cost due to channel failures also increase (Figure 9d). The total average missing cost may increase or decrease (see Figure 9e). However, the total average cost increases with c_m as observed in Figure 9f.

b) $\frac{c_f}{q} \le c_m$*:* In this case, $\frac{c_f}{q} \le c_m + \frac{p_n\beta c_m^2}{2c_a\lambda_n}$ for all the contents. From Theorem 5, the Whittle indices do not depend on c_m in this case. Moreover, following the discussion in Section IV-C, the Whittle index based policy always serves all the contents that are requested. Consequently, for $c_m \ge 1.1$, the average ageing and fetching costs do not vary (Figures 9b and 9c), the average missing cost due to channel



(a) (b)

Fig. 10. Variation in average cost and its components with respect to c_a for $\frac{c_f}{q} \leq c_m$.



(a) (b) (c)

Fig. 11. Variation in the average cost and its components with respect to c_a for $\frac{c_f}{q} > c_m$.

failures increases linearly with c_m (Figure 9d) and the average missing cost due to service denial is 0 (Figure 9a). As a result, both the average missing cost and the overall average cost increase with c_m (see Figures 9a and 9f).

4) Effect of the per Unit Ageing Cost c_a on the Average Costs: We again consider two cases, $\frac{c_f}{q} \leq c_m$ and $\frac{c_f}{q} > c_m$.

a) $\frac{c_f}{q} \leq c_m$: We set $c_f = 1$, $c_m = 2$, $q = 0.9$, $M = 100$ and vary c_a from 0.1 to 15, keeping the other parameters same as in Section VI-B. In this case, the Whittle index based policy always serves the requested contents, i.e., the missing cost is merely due to the channel failures. It fetches a requested content n if its age exceeds τ^{*n} and serves the cached version otherwise (see Section IV-A). Also, from Theorem 2, τ^{*n} decreases as c_a increases. Hence, the proposed policy is more inclined towards fetching fresh versions of requested contents for larger c_a. Consequently, as c_a increases while keeping c_f and c_m fixed, the average missing cost remains unchanged (Figure 10a), the average fetching cost increases (Figure 10b), the average rate at which cached contents are served decreases but the average ageing cost increases (Figure 10a). Since the average missing cost remains unchanged and the average fetching and ageing costs increase with c_a, the overall average cost also increases (Figure 10b).

b) $\frac{c_f}{q} > c_m$: We set $c_f = 1$ and $c_m = 0.8$ and vary c_a from 5 to 25 while keeping the other parameters same as in the previous case. Recall that the Whittle index based policy *does not serve* the requested contents n for which $\frac{c_f}{q} > c_m + \frac{p_n \beta c_m^2}{2 c_a \lambda_n}$ (see Section IV-C). The number of these contents increases as c_a increases, and so, the number of service denials also increases. Therefore, as c_a increases, the average missing cost due to service denials increases (Figure 11a), the average fetching and ageing costs decrease (Figures 11a and 11b), and the average missing cost due to channel failures also decreases (Figure 11b). We observe in Figure 11c that the overall average missing cost and the overall average cost increases with c_a.

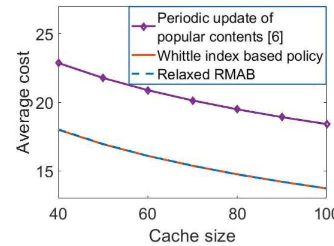


Fig. 12. Performance of Whittle index based policy compared to the existing works [6].

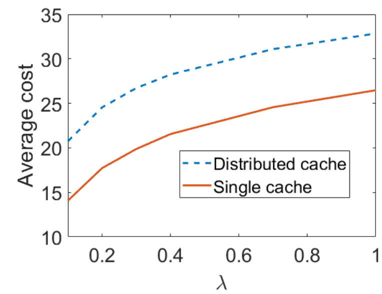


Fig. 13. Average cost vs. λ for single cache framework with Whittle index based policy and for multi-cache framework with load-splitting and distributed caching [13].

C. Comparison to the Policies in [6] and [12]

Abolhassani et al. have considered a similar caching model as ours in [6] and have extended it to allow an unreliable wireless edge between the server and cache in [12]. In contrast, we consider an unreliable wireless channel between the cache and the users. For comparison, we assume reliable channels everywhere. In other words, we consider $q = 1$. The problems and the policies proposed in [6] and [12] become identical under this hypothesis. Since [6], [12] do not consider service denials, we also set the missing cost $c_m = \infty$. Further, we set $c_f = 1$, $c_a = 0.1$, $\beta = 40$, and $p_n \propto \frac{1}{n}$. Since $\frac{c_f}{q} \leq c_m$, the Whittle index based policy always serves the requested contents. We observe in Figure 12 that the Whittle index based policy outperforms the one in [6]. The policies in [6] and [12] are static and do not require any runtime computation. On the other hand, the Whittle index based policy requires $O(M)$ at each decision epoch, where M is the cache size. It must be noted that the policies in [6] and [12] are evidently suboptimal whereas the Whittle index based policy is asymptotically optimal in the regime of large numbers contents and large cache sizes.

D. Single Vs. Distributed Cache

We now compare performances of the proposed Whittle index based policy and the policy in [13] which we described in Section V. Recall that $q = 1$ and $c_m = \infty$ in this case. We set $c_f = 1$, $c_a = 0.1$, $\beta = 40$, $\alpha = 0.9$ and $p_n \propto \frac{1}{n}$. Since $\frac{c_f}{q} \leq c_m$, the Whittle index based policy always serves the requested contents. We vary the update rate λ from 0.1 to 1 and plot the average costs under the Whittle index based policy and the policy in [13] in Figure 13. As discussed in Section V, for sake of comparison, we assume the cache size $M = K^* - 1$ under our single cache framework.

From (22), K^* decreases as λ increases. The increase in λ and the resulting reduction in the cache size increase the overall cost (see Figure 13). We also observe in Figure 13 that the Whittle index based policy performs much better than the policy in [13].

VII. Conclusion

We have considered an optimal content fetching, caching and delivery problem subject to hard cache capacity constraints. We have framed this problem as a restless multi-armed bandit problem. In Theorem 2, we have provided the optimal policy for the single content problem with holding cost. In Theorem 4, we have proved the indexability of the single content problem. In Theorem 5, we have computed the Whittle indices. We have presented the Whittle index based policy to solve the optimal content fetching and caching problem in Section IV-C. We have demonstrated that our proposed policy outperforms the solution provided in [13] and offers almost same performance as the optimal policy.

A potential future direction is to develop a policy where the BS is unaware of the contents' update rates and the request rates. We would also like to extend this study to a network of edge caches each having its own user base.

References

[1] H. Zhou, L. Tang, Q. Huang, and W. Lloyd. (2016). *The Evolution of Advanced Caching in the Facebook CDN*. Accessed: May 14, 2025. [Online]. Available: https://research.fb.com/blog/2016/04/the-evolution-ofadvanced-caching-in-the-facebook-cdn/

[2] K. S. Candan, W.-S. Li, Q. Luo, W.-P. Hsiung, and D. Agrawal, "Enabling dynamic content caching for database-driven web sites," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, May 2001, pp. 532–543.

[3] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "On the age of information with packet deadlines," *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6419–6428, Sep. 2018.

[4] G. Ahani, D. Yuan, and S. Sun, "Optimal scheduling of age-centric caching: Tractability and computation," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2939–2954, Aug. 2022.

[5] R. Talak, S. Karaman, and E. Modiano, "Improving age of information in wireless networks with perfect channel state information," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1765–1778, Aug. 2020.

[6] B. Abolhassani, J. Tadrous, A. Eryilmaz, and E. Yeh, "Fresh caching for dynamic content," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.

[7] R. D. Yates, "The age of gossip in networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 2984–2989.

[8] E. Delfani and N. Pappas, "Version age-optimal cached status updates in a gossiping network with energy harvesting sensor," in *Proc. 21st Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, Aug. 2023, pp. 143–150.

[9] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Achieving freshness in single/multi-user caching of dynamic content over the wireless edge," in *Proc. 18th Int. Symp. Modelling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOPT)*, Jun. 2020, pp. 1–8.

[10] B. Abolhassani, J. Tadrous, A. Eryilmaz, and S. Yüksel, "Optimal push and pull-based edge caching for dynamic content," *IEEE/ACM Trans. Netw.*, vol. 32, no. 4, pp. 2765–2777, Aug. 2024.

[11] A. Koley and C. Singh, "Fresh caching of dynamic contents using restless multi-armed bandits," in *Proc. IEEE 21st Int. Conf. Mobile Ad-Hoc Smart Syst. (MASS)*, Sep. 2024, pp. 238–246.

[12] B. Abolhassani, J. Tadrous, A. Eryilmaz, and E. Yeh, "Fresh caching of dynamic content over the wireless edge," *IEEE/ACM Trans. Netw.*, vol. 30, no. 5, pp. 2315–2327, Oct. 2022.

[13] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Optimal load-splitting and distributed-caching for dynamic content over the wireless edge," *IEEE/ACM Trans. Netw.*, vol. 31, no. 5, pp. 2178–2190, Oct. 2023.

[14] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Single vs distributed edge caching for dynamic content," *IEEE/ACM Trans. Netw.*, vol. 30, no. 2, pp. 669–682, Apr. 2022.

[15] G. Chen and S. C. Liew, "An index policy for minimizing the uncertainty-of-information of Markov sources," *IEEE Trans. Inf. Theory*, vol. 70, no. 1, pp. 698–721, Jan. 2024.

[16] S. Zhang, L. Wang, H. Luo, X. Ma, and S. Zhou, "AoI-delay tradeoff in mobile edge caching with freshness-aware content refreshing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5329–5342, Aug. 2021.

[17] K. D. Glazebrook, H. M. Mitchell, and P. S. Ansell, "Index policies for the maintenance of a collection of machines by a set of repairmen," *Eur. J. Oper. Res.*, vol. 165, no. 1, pp. 267–284, Aug. 2005.

[18] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5547–5567, Nov. 2010.

[19] V. Tripathi and E. Modiano, "A whittle index approach to minimizing functions of age of information," *IEEE/ACM Trans. Netw.*, vol. 32, no. 6, pp. 5144–5158, Dec. 2024.

[20] N. Akbarzadeh and A. Mahajan, "Conditions for indexability of restless bandits and an algorithm to compute whittle index," *Adv. Appl. Probab.*, vol. 54, no. 4, pp. 1164–1192, Dec. 2022.

[21] N. Gast, B. Gaujal, and K. Khun, "Testing indexability and computing whittle and Gittins index in subcubic time," *Math. Methods Oper. Res.*, vol. 97, no. 3, pp. 391–436, Jun. 2023.

[22] T. Stahlbuhk, B. Shrader, and E. Modiano, "Throughput maximization in uncooperative spectrum sharing networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 6, pp. 2517–2530, Dec. 2020.

[23] G. Xiong, S. Wang, J. Li, and R. Singh, "Whittle index-based Q-learning for wireless edge caching with linear function approximation," *IEEE/ACM Trans. Netw.*, vol. 32, no. 5, pp. 4286–4301, Oct. 2024.

[24] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Mar. 1999, pp. 126–134.

[25] P. Whittle, "Restless bandits: Activity allocation in a changing world," *J. Appl. Probab.*, vol. 25, no. A, pp. 287–298, 1988.

[26] Q. Zhao, *Multi-Armed Bandits: Theory and Applications to Online Learning in Networks*. Cham, Switzerland: Springer Nature, 2022.

[27] S. Aalto, P. Lassila, and I. Taboada, "Whittle index approach to opportunistic scheduling with partial channel information," *Perform. Eval.*, vol. 136, Dec. 2019, Art. no. 102052.

[28] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2, 3rd ed., Belmont, MA, USA: Athena Scientific, 2007.

[29] A. Koley and C. Singh, "Fresh caching of dynamic contents using restless multi-armed bandits," 2024, *arXiv:2404.12468*.

[30] U. Ayesta, M. K. Gupta, and I. M. Verloop, "On the computation of Whittle's index for Markovian restless bandits," *Math. Methods Oper. Res.*, vol. 93, no. 1, pp. 179–208, 2021.

[31] T. Ahrendt, "Fast computations of the exponential function," in *Proc. Annu. Symp. Theor. Aspects Comput. Sci.*, 1999, pp. 302–312.

[32] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 450. Belmont, MA, USA: Athena Scientific, 1995.

Ankita Koley received the B.Tech. degree in electrical engineering from the National Institute of Technology, Durgapur, in 2018. She is currently pursuing the Ph.D. degree with the Department of Electronic Systems Engineering, Indian Institute of Science, Bengaluru. Her research interests include communication networks, edge caching, edge computing, and optimization theory.

Chandramani Singh (Member, IEEE) received the M.E. and Ph.D. degrees in electrical communication engineering from Indian Institute of Science, Bengaluru, India, in 2005 and 2012, respectively. He was a Research Engineer with INRIA Rocquencourt from 2012 to 2013, and a Post-Doctoral Research Associate with CSL, University of Illinois Urbana-Champaign from 2013 to 2014. He is currently an Associate Professor with the Department of ESE, Indian Institute of Science, Bengaluru. His research interests include communication networks, stochastic control and learning, and game theory.