

# Fresh Caching of Dynamic Contents using Restless Multi-armed Bandits

Ankita Koley, Chandramani Singh

Department of Electronic Systems Engineering

Indian Institute of Science

Bangalore 560012, India

Email:{ankitakoley, chandra}@iisc.ac.in

**Abstract**—We consider a dynamic content caching problem wherein the contents get updated at a central server, and local copies of a subset of contents are cached at a local cache associated with a Base station (BS). When a content request arrives, based on whether the content is in the local cache, the BS can decide whether to fetch the content from the central server or serve the cached version from the local cache. Fetching a content incurs a fixed fetching cost, and serving the cached version incurs an ageing cost proportional to the age-of-version (AoV) of the content. The BS has only partial information regarding AoVs of the contents. We formulate an optimal content fetching and caching problem to minimize the average cost subject to cache capacity constraints. The problem suffers from the curse of dimensionality and is provably hard to solve. We formulate this problem as a continuous time restless multi-armed bandit process (RMAB), where a single content problem of the corresponding RMAB is a partially observable Markov decision process. We reformulate the single content problem as a semi-Markov decision process, prove indexability, and provide a Whittle index based solution to this problem. Finally, we compare the performance with recent work and show that our proposed policy is optimal via simulations.

## I. INTRODUCTION

Over the past few years, online social networks (OSNs) like Facebook, Instagram, LinkedIn, and YouTube have become viral platforms for users to interact, communicate, and share content over the internet. The increasing popularity has attracted many new users, resulting in a huge volume of content being shared across these platforms.

The OSNs' Content Distribution Networks (CDNs) deploy caches in various geographical locations nearer to the users along with the central server cache to ensure the timely delivery of content; for example, the Facebook content distribution Network (FBCDN) uses several layers of caches along with backend cache [1]. Along with low latency, caching reduces the backhaul traffic, reducing congestion, specifically during peak hours. Depending upon users' interaction, content relevance, location, etc., the content dynamics change, for example, Facebook's news feed or YouTube's recommendation system. As the dynamic contents get updated at the central server, these need to be replaced at the local cache of the CDNs. The server holds the most relevant version of the content for most applications and web pages [2].

This work was supported jointly by Centre for Network Intelligence, Indian Institute of Science (IISc), a CISCO CSR initiative and Aircel TCoE project 39010C.

The cached contents at the local cache may lose their relevance to users over time as they get updated at the central server. Upon receiving a request, whether the local cache will serve the cached content or fetch a fresh version depends upon *freshness* of the content. The *freshness* of content is measured by age-of-version (AoV), a metric proposed by Abolhassani et al. [3]. AoV of content is the number of updates in the central server since that particular content is being fetched. The AoV of content depends on the age of the content and the frequency at which the content gets updated at the central server. Once a content is cached, it needs to be replaced with newer versions depending upon the AoV of the content. Otherwise, the content will become stale, resulting in an aging cost depending upon the AoV. Conversely, fetching a fresh version of the content will incur a fetching cost. Hence, designing caching policies for dynamic contents poses the following challenges:

- 1) *Unknown Content dynamics*: Since the contents get updated at the central server, the AoV may not be known to the local cache unless the content is fetched.
- 2) *Constrained cache capacity*: Local cache has smaller capacity in comparison to the central server. Hence, after fetching fresh version of the content it needs to decide whether to cache it or not based on the the cached content and cache capacity.
- 3) *Dynamic requests*: Content requests vary dynamically depending on popularity, location, and many more factors.

Based on these factors, to minimize the cost, caching policies must carefully decide when to fetch content and whether to cache the fetched content by replacing one of the cached contents or discard the one, keeping the cached contents as they are. We aim to design a caching policy to minimize the average fetching and ageing costs subject to the caching constraint. This cost minimization problem falls in the restless multi-armed bandit process (RMAB) class, where we refer to each content as an arm. The single content problem of the corresponding RMAB evolves as a partially observable Markov decision process (POMDP). We further reformulate the POMDP as a semi-Markov decision process, retaining all essential information. Finally, we propose a Whittle index-based policy to solve the problem.

## A. Related Work

Kam et al. [4] present a framework that minimizes the cache miss rate, considering content requests affected by information

freshness and popularity. However, the assumptions that there are fewer requests for packets for the content with higher age and that information freshness depends solely on the age of the content rather than the number of updates may not align well with the dynamics of today's internet. This model is further generalized by Ahanai et al. [5] introducing utility functions that depend upon age and the popularity. These Age-of-information (AoI) driven models overlook the impact on freshness due to updation of contents at the central server. Yates [6] introduces version AoI, a very similar metric to AoV, and studies the minimization of average version AoI in a gossiping network. Version AoI or AoV is a more suitable metric to measure freshness as it measures the number of updates since a fresh version is fetched. However, dealing with AoV can be challenging for most of the systems as the local cache may not be aware of the updates at the central server. Abolhassani et al. [7] pose two optimization frameworks. In the single-user scenario, the user checks for an update at the central server, incurring a check cost, and then updates its cache, resulting in a cache cost. In this case, the user can store more than one content. In this model, when a request for cached content comes, the user must serve the version it currently has and the user fetches only if there is a cache miss. In the multi-user scenario, each user updates its cache for free when another user requests the item via broadcast. Each user has a cache capacity of precisely one, potentially resulting in multiple copies of the same content.

We consider a more flexible framework similar to [3], i.e., when a request for cached content comes, the local cache can serve the version it has or request a fresh version from the server. Furthermore, in our model, the local cache does not employ a cache check; hence, it does not know when it gets updated at the server. As mentioned in [3], this problem falls under the scope of a partially observable Markov decision process (POMDP) average cost problem, hard to solve. Hence, the hard cache constraint has been replaced with a probabilistic constraint to solve the problem. Further, a more flexible choice of average cache capacity constraint has been considered to satisfy the probabilistic constraint. Their solution suggests that it is enough to cache most popular items.

In a recent work, Abolhassani et al. [8] consider a combined push and pull based caching policy, where in push based policy the central server takes the decision to update the content in the local cache, exploiting the knowledge of exact number updates of a content and in pull based caching policy the decisions are taken at the local cache exploiting the exact knowledge of request arrival. Under the assumption that once a content is stored in the local cache, it will never be discarded; push based caching policy is applied on a subset of the cached contents and pull based caching is applied for the other contents. They show that the push and pull based caching is optimal for the cached contents. However, for the uncached content, whenever there is a request there will be constant fetching cost every time. Hence, this problem needs further investigation on how to apply a push based caching when there is a provision to discard the content. However, in our work we relax the above constraint, i.e., a cached content can be discarded followed by

storing a new content and focus on pull based caching, i.e., the local cache takes the caching decisions.

In another recent work, Abolhassani et al. [9] consider a pull based caching framework with average cache constraint and study a model based and model free learning. The solution based on average cache constraint is not practical, since the actual cache capacity might be exceeded while implementing the caching policy. In comparison with [3] where the authors consider dynamic caching problem with a probabilistic constraint, our work focuses on developing an efficient approach to resolving the dynamic caching problem subject to a hard cache constraint.

## B. Our Contribution

- 1) We pose an optimal content fetching and caching problem to minimize the time average cost subject to local cache capacity constraints. This problem falls under the scope of continuous time restless multi-armed bandit process (RMAB), where the single content problem evolves as a POMDP.
- 2) We reformulate the single content POMDP as a semi-Markov decision process and establish the indexability for each content. We further obtain a closed form expressions of the Whittle indices and propose the Whittle index based policy to solve the multi-content problem.
- 3) We implement the proposed Whittle index-based policy via simulations and show that the Whittle Index based policy outperforms the policy proposed by Abolhassani et al. [3, Theorem 2]. We also show that the Whittle index based policy offers the same performance as the optimal policy.

## II. SYSTEM MODEL

Let us consider a communication network with a central server, a Base Station (BS) associated with a local 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. It can fetch and store up to  $M$  contents in the local cache and can locally serve these to the users upon request (Figure 1).

*Content dynamics:* All the  $N$  contents are updated according to independent Poisson processes with  $\lambda_n$  being the update rate of the  $n^{th}$  content. The central server always hosts the latest version of the contents. Setting  $\lambda_n = 0$  for all  $n$  yields the special case of *static* contents.

*Request dynamics:* The aggregate request process of the end users is a Poisson process with rate  $\beta$ . 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 [10]. Under the proposed request dynamics,  $n^{th}$  content's request rate is a Poisson process with rate  $p_n\beta$ . Let  $\mathcal{C}(t) \subset \{1, \dots, N\}$  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

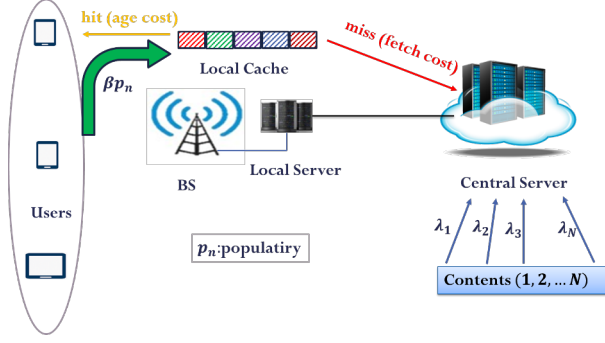


Fig. 1: Fresh caching of dynamic content

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  $t$ .<sup>1</sup> 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.

*Content fetching and ageing costs:* If a content, say Content  $n$ , is requested at time  $t$ , one of the following scenarios may occur.

- Content  $n$  is found at the local cache. In this case, the cached copy can be served. 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 update. Alternatively, the latest version of Content  $n$  can be fetched and served, incurring a constant *fetching cost*  $c_f$ . The newly fetched copy replaces the existing one in the cache.
- Content  $n$  is not found at the local cache. It is then fetched at cost  $c_f$  and served. The fetched content either can replace an exiting content in the cache or can be discarded after serving.

Let  $A(T)$  denote the total number of requests until time  $T$ . Let  $t_k \geq 0$  and  $e_k \in \{1, \dots, N\}$  denote the  $k^{th}$  request epoch and identity of the requested content at  $t_k$ , respectively. The BS does not have access to the information about the updates happening in the central server. However, the BS knows the rate ( $\lambda_n$ ) at which the contents are getting updated. Further, let  $a_k^n$  denote the action at  $t_k$  vis a vis Content  $n \in \mathcal{C}(t_k) \cup \{e_k\}$ ;  $a_k^n \in \{0, 1\}$  if  $n \in \{e_k\} \cap \mathcal{C}(t_k)$ ,  $a_k^n \in \{1, 3\}$  if  $n \in \{e_k\} \setminus \mathcal{C}(t_k)$ , and  $a_k^n \in \{0, 2\}$  otherwise. Hence, we define the action

TABLE I: Annotations of the actions

Actions	Contents	Annotations
0	$n \in \mathcal{C}(t_k) \cap \{e_k\}$ $n \in \mathcal{C}(t_k) \setminus \{e_k\}$	serve the cached copy and keep keep
1	$n = e_k$	fetch, serve and cache
2	$n \in \mathcal{C}(t_k) \setminus \{e_k\}$	discard
3	$n = e_k$	fetch, serve and discard

set as  $\mathcal{A} := \{a \in \{0, 1, 2, 3\}^N : \sum_{n=1}^N \mathbb{1}_{\{a^n \in \{0, 1\}\}} = M\}$

<sup>1</sup>We use the phrases “ $n^{th}$  content” and “Content  $n$ ” interchangeably.

### A. The Optimal Content Fetching and Caching Problem

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

$$\inf_{\pi \in \Pi} \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E}_{\pi} \left[ \sum_{k=1}^{A(T)} \left( \mathbb{1}_{\{a_k^{e_k}=0\}} c_a \nu_{e_k}(t_k) + \mathbb{1}_{\{a_k^{e_k} \in \{1, 3\}\}} c_f \right) \right] \quad (1)$$

$$\text{s. t. } \sum_{n=1}^N \mathbb{1}_{\{a_k^n \in \{0, 1\}\}} = M, \forall k \geq 1 \quad (2)$$

where  $\Pi$  is a set of feasible policy.

*Discussion:* The optimal content fetching and caching problem (1) is a Markov decision process (MDP) with an exponential state space and action space in  $M + 1$ . One can solve this problem using dynamic programming by restricting the states to be finite, but the solution suffers from the curse of dimensionality. However, the MDP in (1) falls in the class of continuous time restless multi-armed bandit process (RMAB) and Whittle index policy [11] is a good heuristic solution to RMAB problems.

*Whittle Index Policy:* We consider the following relaxed constraint instead of the hard constraint (2):

$$\lim_{T \rightarrow \infty} \frac{1}{\beta T} \mathbb{E}_{\pi} \left[ \sum_k \sum_{n=1}^N \mathbb{1}_{\{a_k^n \in \{0, 1\}\}} \right] = M \quad (3)$$

Hence, we can write the Lagrangian of the problem (1) subject to the relaxed constraint (3) with multiplier  $C_h$  as follows:

$$\begin{aligned} & \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E}_{\pi} \left[ \sum_{k=1}^{A(T)} \left( \mathbb{1}_{\{a_k^{e_k}=0\}} c_a \nu_{e_k}(t_k) + \mathbb{1}_{\{a_k^{e_k} \in \{1, 3\}\}} c_f \right) \right] \\ & + C_h \left( \lim_{T \rightarrow \infty} \frac{1}{\beta T} \mathbb{E}_{\pi} \left[ \sum_k \sum_{n=1}^N \mathbb{1}_{\{a_k^n \in \{0, 1\}\}} \right] - M \right) \\ & = \sum_{n=1}^N \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E}_{\pi} \left[ \sum_k \left( \mathbb{1}_{\{a_k^n=0\}} c_a \nu_n(t_k) + \mathbb{1}_{\{a_k^n \in \{1, 3\}\}} c_f \right) \right. \\ & \quad \left. \mathbb{1}_{\{e_k=n\}} + \frac{C_h}{\beta} \mathbb{1}_{\{a_k^n \in \{0, 1\}\}} \right] - C_h M \\ & = \sum_{n=1}^N V_{\pi}^n(C_h) - C_h M \end{aligned} \quad (4)$$

where  $V_{\pi}^n(C_h)$  is the expected average cost of Content  $n$  under policy  $\pi$  where a holding cost of  $C_h$  per unit time is incurred for taking action 0 or 1, i.e., keeping or caching the Content  $n$ , respectively. The above problem (4) can be decoupled into  $n$  single content problems except the common cost part,  $C_h M$  which is independent of the policy  $\pi$ . We define  $V_{\pi}^{\pi_n(C_h)} := \inf_{\pi \in \Pi} V_{\pi}^n(C_h)$  to be the value of the single content problem,  $V_{\pi}^n(C_h)$  under the optimal policy  $\pi_n(C_h)$ . Let the optimal solution to the relaxed problem (4) be  $\bar{V} = \max_{C_h} (\sum_{n=1}^N V_{\pi_n(C_h)}^n - C_h M)$ . 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 formally in sections III-A and IV, respectively. Indexability of a content is a property that holds if the optimal action of the single content problem  $\notin \{0, 1\}$ , i.e., not to keep or cache the content for a given holding cost  $C_h$ , then it is not optimal to hold that content for a higher holding cost,  $C'_h > C_h$ . Whittle index for a content is the minimum value of  $C_h$  for which it is equally attractive to keep or discard the Content  $n$ . We further denote  $\hat{V}$  and  $V_W$  to be the value of the optimal policy and the Whittle index policy of the original problem, i.e., (1) under the hard constraint (2). Then, it is easy to show that  $\hat{V} \leq \hat{V} \leq V_W$  [12]. According to Whittle's conjecture [11] as the number of contents,  $N$  grows to infinity and the buffer 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 and caching problem (1) subject to (3).

We note that for  $C_h < 0$  the optimal policy for the single content problem is to keep, i.e.,  $\{0, 1\}$ . More precisely, when the requested content is not in the cache, the optimal action is 1 for the requested content's single content problem and the optimal action is 0 for other contents' single content problems. Hence, similar to [13], we will also consider  $C_h \geq 0$  for our analysis. Going forward we will consider the single content problem for  $C_h \geq 0$ . We first show that the single content problem is indexable and then we compute the Whittle indices for each content.

*Remark 1:* Abolhassani et al. [3] provides an optimal solution by replacing (2) with a probabilistic constraint, i.e., the steady state fraction of time the items are held in the cache does not exceed the cache capacity with high probability. To satisfy the probabilistic constraint, the authors further consider an average cache constraint and using KKT conditions they provide an optimal solution, i.e., which items to cache and for how long these should be cached. However, the constraint on average cache occupancy,  $\bar{M} = Me^{-v}$ , where  $v \in \mathbb{N}$  and  $v$  satisfy the equation in Proposition 1 of [3] to ensure that the probabilistic constraint is satisfied. The caching strategy is to cache most popular items and hold them for the times given by the solution of while the average cache occupancy is  $\bar{M}$  to ensure that that number of popular items does not exceed the actual cache capacity. The caching strategy does not cache the unpopular items at all even if the cache is not full, for each request of these unpopular items, a cost of  $c_f$  incurs. This leads to under utilizing the cache capacity that further increases the average cost. However, Whittle index policy utilizes the full cache capacity and we demonstrate in section V that it performs better than the solution provided by the authors in [3]. Moreover, we also show that the cost obtained by the Whittle index policy is very close to the optimal value of the relaxed problem, i.e., the optimal content fetching and caching problem (1) subject to (3). This implies that the Whittle index policy offers almost the same performance as the optimal policy for the optimal content fetching and caching problem

with hard cache constraints.

### III. SINGLE CONTENT PROBLEM

In this section, we discuss the single content problem with holding cost. Let  $C_h \geq 0$  be the holding cost of a content per unit of time. At each epoch  $t_k$  there is a request for the content with probability  $p$ . Let  $A(T)$  be the number of decision epoch until time  $T$ . If there is a request for the content at epoch  $k$  and the content is in the cache then the following actions can be taken:

- (a) serve the cached copy and keep
- (b) fetch serve and cache
- (c) fetch serve and discard
- (d) serve the cached copy and discard.

Recall that from the Table I the first three actions (a), (b) and (c) are 0, 1 and 3, respectively. We denote the last action (d) to be 2. We note that we did not include the action, *serve the cached copy and discard* in action 2 of the multi content problem (1) with the constraint on the cache capacity (2) in section II-A. In the multi content problem there is no cost associated with keeping or caching a content in the cache, hence including the action, *serve the cached copy and discard* in action 2 does not change the optimal solution and policy of the problem. We aim to minimize the following average cost for single content:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{k=1}^{A(T)} \mathbb{1}_{\{a_k \in \{0,2\}\}} c_a \nu(t_k) + \mathbb{1}_{\{a_k \in \{1,3\}\}} c_f + \mathbb{1}_{\{a_k \in \{0,1\}\}} (t_{k+1} - t_k) C_h \right] \quad (5)$$

Instead of considering the (5) we can consider the following equivalent objective [14] for minimization:

$$\inf \lim_{T \rightarrow \infty} \frac{1}{\mathbb{E}[A(T)]} \mathbb{E} \left[ \sum_{k=1}^{A(T)} \mathbb{1}_{\{a_k \in \{0,2\}\}} c_a \nu(t_k) + \mathbb{1}_{\{a_k \in \{1,3\}\}} c_f + \mathbb{1}_{\{a_k \in \{0,1\}\}} (t_{k+1} - t_k) C_h \right] \quad (6)$$

The above minimization problem is a partially observed Markov decision process (POMDP) where the AoV at epoch  $k$ ,  $v(t_k)$  of the content 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. We denote this quantity as  $\tau_k := t_k - \max\{t_l : l < k : a_l = 1\}$ . The expected AoV at epoch  $k$ ,  $\mathbb{E}[\nu(t_k)] = \lambda \tau_k$ . Hence, we reformulate the above POMDP problem as a semi-Markov decision process considering  $\tau_k$  instead of  $v(t_k)$  as a part of the state. Let  $b_k$  and  $y_k$  be two indicator variables for the content. In each request epoch  $k$ , if there is a request for the content, then  $b_k = 1$  and 0, otherwise. Similarly, if the content is found in the cache, 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,  $\mathcal{S} := \{(\tau, 1, 1), (\tau, 1, 0), (0, 1), (0, 0), \tau \geq 0\}$ . The state at time  $t$  are constant in between two consecutive epochs for  $t_k \leq t < t_{k+1}$ . The time interval between  $k^{th}$  and  $(k+1)^{th}$

request epoch is exponentially distributed with parameter  $\beta$ . Given,  $s_k = (\tau, 1, b)$ , in the  $(k+1)^{th}$  epoch the states are updated as follows:

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

where  $\Delta\tau \sim \exp(\beta)$  and  $b' \sim \text{Bernoulli}(p)$ . Recall from Table I that action 1 is applicable only when the state is  $(\tau, 1, 1)$ . Given,  $s_k = (0, 1)$ , in the  $(k+1)^{th}$  epoch the states are updated as follows:

$$s_{k+1} = \begin{cases} (\Delta\tau, 1, b'), & \text{if } a_k = 1 \\ (0, b'), & \text{if } a_k = 3 \end{cases} \quad (8)$$

Recall from Table I that action 0 and 2 are 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 of each content is given  $c(s, a) = \begin{cases} c_a \lambda \tau \mathbb{1}_{\{a \in \{0, 2\}\}} + c_f \mathbb{1}_{\{a \in \{1, 3\}\}} + \frac{C_h}{\beta} \mathbb{1}_{\{a \in \{0, 1\}\}}, & \text{if } s = (\tau, 1, 1) \\ c_f \mathbb{1}_{\{a \in \{1, 3\}\}} + \frac{C_h}{\beta} \mathbb{1}_{\{a=1\}}, & \text{if } s = (0, 1) \\ \frac{C_h}{\beta} \mathbb{1}_{\{a=0\}}, & \text{if } s = (\tau, 1, 0) \\ 0, & \text{if } s = (0, 0) \end{cases}$

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

$$J_\pi(s) = \lim_{T \rightarrow \infty} \frac{1}{\mathbb{E}_\pi[A(T)]} \mathbb{E}_\pi \left[ \sum_{k=1}^{A(T)} c(s_k, a_k) | s_0 = s \right] \quad (9)$$

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

where  $a_k = \mu_k(s_k)$  and  $J_*(s)$  is the solution under optimal policy. It can be shown from [14] that  $J_\pi(s)$  and  $J_*(s)$  are independent of  $s$  since the embedded discrete time Markov chains of the problems (9) and (10) have single recurrent classes. Moreover, Bellman's equation for the semi-Markov problem is similar to the discrete-time problems [14, Chapter 5, Section 5.3]. Suppose,  $h(s)$  and  $\theta$  are the relative value function of state  $s$  and optimal value, respectively. Let us denote  $L_r(\tau) = \int_r^\infty \beta e^{-\beta t} (ph(t + \tau, 1, 1) + (1-p)h(t + \tau, 1, 0)) dt$ . Then Bellman's equations from each state are as follows [14, Chapter 5, Section 5.3]:

$$h(\tau, 1, 1) = \min \left\{ c_a \lambda \tau + \frac{C_h - \theta}{\beta} + L_0(\tau), c_f + \frac{C_h - \theta}{\beta} + L_0(0), c_a \lambda \tau - \frac{\theta}{\beta} + ph(0, 1) + (1-p)h(0, 0), c_f - \frac{\theta}{\beta} + ph(0, 1) + (1-p)h(0, 0) \right\} \quad (11)$$

$$h(\tau, 1, 0) = \min \left\{ \frac{C_h - \theta}{\beta} + L_0(\tau), -\frac{\theta}{\beta} + ph(0, 1) + (1-p)h(0, 0) \right\} \quad (12)$$

$$h(0, 1) = \min \left\{ c_f + \frac{C_h - \theta}{\beta} + L_0(0), \right.$$

$$\left. c_f - \frac{\theta}{\beta} + ph(0, 1) + (1-p)h(0, 0) \right\} \quad (13)$$

$$h(0, 0) = -\frac{\theta}{\beta} + ph(0, 1) + (1-p)h(0, 0) \Rightarrow h(0, 0) = -\frac{\theta}{p\beta} + h(0, 1) \quad (14)$$

Using (14) we can rewrite the equations for  $h(0, 1)$ ,  $h(\tau, 1, 0)$  and  $h(\tau, 1, 1)$  as follows:

$$h(\tau, 1, 1) = \min \left\{ c_a \lambda \tau + \frac{C_h - \theta}{\beta} + L_0(\tau), c_f + \frac{C_h - \theta}{\beta} + L_0(0), c_a \lambda \tau + h(0, 0), c_f + h(0, 0) \right\} \quad (15)$$

$$h(\tau, 1, 0) = \min \left\{ \frac{C_h - \theta}{\beta} + L_0(\tau), h(0, 0) \right\} \quad (16)$$

$$h(0, 1) = \min \left\{ c_f + \frac{C_h - \theta}{\beta} + L_0(0), c_f + h(0, 0) \right\} \quad (17)$$

**Lemma 1:**

- (a) Both  $h(\tau, 1, 1)$  and  $h(\tau, 1, 0)$  are non-decreasing in  $\tau$ .
- (b) For a given  $r \geq 0$ ,  $L_r(\tau)$  is non-decreasing in  $\tau$ .

*Proof:*

- (a) We use induction to prove the claim. The detailed proof can be found in our technical report [15, Appendix A].
- (b) Since  $h(\tau, 1, 1)$  and  $h(\tau, 1, 0)$  are non-decreasing in  $\tau$ , the integrand in  $L_r(\tau)$  is also non-decreasing in  $\tau$ . Hence  $L_r(\tau)$  is non-decreasing in  $\tau$ .

**Theorem 2:** The optimal policy  $\pi^*$  depends on the holding cost  $C_h$  and is as shown in Table II, where  $I = p\beta c_f - pc_a \lambda (1 - e^{-\beta \tau^0})$ ,  $\tau^0 = \frac{c_f}{c_a \lambda}$ ,  $\tau^* = -\frac{1}{p\beta} + \sqrt{\left(\frac{1}{p\beta}\right)^2 + \frac{2c_f}{p\beta c_a \lambda}}$  and  $(\bar{\tau}, \tilde{\tau})$  are the solutions to the following two equations:

$$c_a \lambda p \beta (\bar{\tau} \tilde{\tau} - \frac{\bar{\tau}^2}{2}) - C_h \bar{\tau} + c_a \lambda \bar{\tau} - c_f = 0 \quad (18)$$

$$\beta (\tilde{\tau} - \bar{\tau}) + e^{-\beta(\tilde{\tau} - \bar{\tau})} - 1 - \frac{C_h}{p c_a \lambda} = 0 \quad (19)$$

Additionally,  $0 \leq \bar{\tau} < \tau^* < \tilde{\tau} \leq \tau^0$ .

TABLE II: State-wise optimal actions for different values of  $C_h$

Holding cost ( $C_h$ )	Optimal policy ( $\pi^*$ )			Optimal cost ( $\theta$ )
	$\pi^*(\tau, 1, 1)$	$\pi^*(\tau, 1, 0)$	$\pi^*(0, 1)$	
$C_h = 0$	0 for $\tau \leq \tau^*$ 1 for $\tau > \tau^*$	0	1	$p\beta c_a \lambda \tau^*$
$0 < C_h < I$	0 for $\tau \leq \bar{\tau}$ 2 for $\bar{\tau} \leq \tau \leq \tilde{\tau}$ 1 for $\tau > \tilde{\tau}$	0 for $\tau \leq \bar{\tau}$ 2 for $\tau > \bar{\tau}$	1	$p\beta c_a \lambda \tilde{\tau}$
$C_h > I$	2 for $\tau \leq \tau^0$ 3 for $\tau > \tau^0$	2	3	$p\beta c_f$

**Remark 2:** 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 ( $\tau$ ). Moreover, it provides the closed forms of thresholds for different values of  $C_h$  in Table II.

*Proof:* We outline the proof of this theorem here, the detailed proof can be found in our technical report [15, Appendix B].

(a) We consider three disjoint cases conditioned on the values of  $h(\tau, 1, 1)$  and  $h(\tau, 1, 0)$  as follows:

I.  $\frac{C_h - \theta}{\beta} + L_0(0) \geq h(0, 0)$ .

II. Since  $h(\tau, 1, 1)$  and  $h(\tau, 1, 0)$  are non-decreasing in  $\tau$ , the complement of case 1 is there exists a  $\bar{\tau} > 0$  such that  $\bar{\tau} = \min \left\{ \tau > 0 : \frac{C_h - \theta}{\beta} + L_0(\tau) \geq h(0, 0) \right\}$ . Since  $L_0(\tau)$  is non-decreasing in  $\tau$ ,  $c_a \lambda \tau + L_0(\tau)$  is strictly increasing in  $\tau$ . Since,  $c_f + L_0(0)$  is a constant and  $c_a \lambda \tau + L_0(\tau) < c_f + L_0(0)$  at  $\tau = 0$ , there exists a  $\tau^* > 0$  such that  $c_a \lambda \tau^* + L_0(\tau^*) = c_f + L_0(0)$ . Hence,  $\tau^* = \min \left\{ \tau \geq 0 : c_a \lambda \tau + L_0(\tau) = c_f + L_0(0) \right\}$ . There could be two cases  $\tau^* \leq \bar{\tau}$  or  $\tau^* \geq \bar{\tau}$ . We consider  $\tau^* \leq \bar{\tau}$  in this case and  $\tau^* > \bar{\tau}$  in the subsequent case.

III. There exists a  $\bar{\tau} > 0$  such that  $\bar{\tau} = \min \left\{ \tau > 0 : \frac{C_h - \theta}{\beta} + L_0(\tau) \geq h(0, 0) \right\}$  and  $\tau^* > \bar{\tau}$ . In this case, for  $\tau \leq \bar{\tau}$ ,  $C_a \lambda \tau + \frac{C_h - \theta}{\beta} + L_0(\tau) \leq C_a \lambda \tau + h(0, 0)$  and  $\tau = \bar{\tau}$ ,

$$C_a \lambda \tau + \frac{C_h - \theta}{\beta} + L_0(\tau) = C_a \lambda \tau + h(0, 0) \quad (20)$$

Since  $L_0(\tau)$  is non-decreasing in  $\tau$ , both R.H.S and L.H.S in (20) are increasing in  $\tau$ . The slope of R.H.S is  $C_a \lambda$  and the slope of L.H.S is at least  $C_a \lambda$ , i.e., the slope of L.H.S is greater than or equal to that of R.H.S. Since at  $\tau^*$ , the L.H.S exceeds  $c_f + \frac{C_h - \theta}{\beta} + L_0(0)$ , it is obvious that there exists a  $\tilde{\tau} \geq \tau^*$  such that, R.H.S exceeds  $c_f + \frac{C_h - \theta}{\beta} + L_0(0)$ . This implies that there exists a  $\tilde{\tau} > \bar{\tau}$  such that  $\tilde{\tau} = \min \left\{ \tau > \bar{\tau} : c_a \lambda \tau + h(0, 0) = c_f + \frac{C_h - \theta}{\beta} + L_0(0) \right\}$ . Now, we will show that suppose  $\exists \tilde{\tau}$  as above, then  $\tau^* > \bar{\tau}$ . Since at  $\tilde{\tau}$  R.H.S exceeds  $c_f + \frac{C_h - \theta}{\beta} + L_0(0)$  and slope of R.H.S is no more than the L.H.S, the L.H.S exceeds  $c_f + \frac{C_h - \theta}{\beta} + L_0(0)$  at a value of  $\tau$  no more than  $\bar{\tau}$ . Hence,  $\tau^* > \bar{\tau}$ . Hence, we consider the last case as follows: there exists  $\bar{\tau} > 0$  such that  $\bar{\tau} = \min \left\{ \tau > 0 : \frac{C_h - \theta}{\beta} + L_0(\tau) \geq h(0, 0) \right\}$  and there exists a  $\tilde{\tau} > \bar{\tau}$  such that  $\tilde{\tau} = \min \left\{ \tau > \bar{\tau} : c_a \lambda \tau + h(0, 0) = c_f + \frac{C_h - \theta}{\beta} + L_0(0) \right\}$ .

(b) We derive the optimal actions in each of the cases. Furthermore, we show that Case I holds iff  $C_h > I$ , Case II holds iff  $C_h = 0$ , and Case III holds iff  $0 < C_h \leq I$ . This also implies that the above mentioned three cases are mutually exclusive and exhaustive.

(c) Recall  $\tau^0 = \frac{c_f}{c_a \lambda}$  theorem 2. Hence, from case III.  $c_a \lambda \tau^0 + h(0, 0) = c_f + h(0, 0) \geq c_f + \frac{C_h - \theta}{\beta} + L_0(0)$ . Hence, by the definition of  $\tilde{\tau}$ ,  $\tilde{\tau} \leq \tau^0$ . Hence, from the Lemma 3  $\bar{\tau} \leq \tau^* \leq \tilde{\tau} \leq \tau^0$ .

*Lemma 3:*

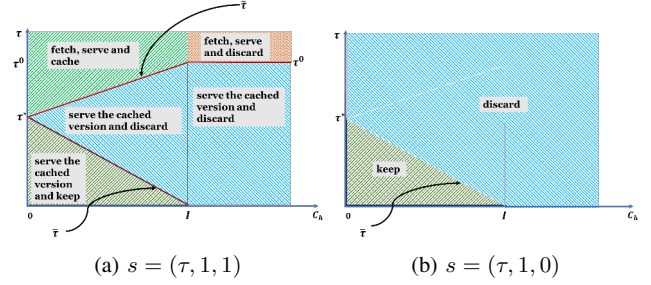


Fig. 2: Optimal policy structure with respect to  $C_h$  for the state  $(\tau, 1, 1)$  and  $(\tau, 1, 0)$  where  $\tau^*$ ,  $\tau^0$  and  $I$  are as in Theorem 2.

- 1) (19) 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  $0 < C_h \leq I$  respectively. Furthermore,  $\bar{\tau} \leq \tau^* \leq \tilde{\tau}$ .

*Proof:* See *Proof of Lemma 3* in our technical report [15, Appendix E].

*Remark 3:* In Figures 2a and 2b, we plot the threshold values of  $\tau$  as we vary  $C_h$  and indicate various regions depending upon optimal actions for the states  $\{(\tau, 1, 1) : \tau \geq 0\}$  and  $\{(\tau, 1, 0) : \tau \geq 0\}$ , respectively. We describe the actions for different states as follows:

- (a) When the state of the content is  $(\tau, 1, 1)$ , given  $C_h = 0$ , the optimal action is to serve the cached version and keep for  $\tau \leq \tau^*$  and fetch, serve, and keep for  $\tau > \tau^*$ . For  $0 < C_h < I$ , the optimal action is to serve the cached version and keep for  $\tau \leq \bar{\tau}$ , serve the cached version and discard for  $\bar{\tau} < \tau \leq \tilde{\tau}$ , and fetch, serve and cache for  $\tau > \tilde{\tau}$ . For  $C_h \geq I$ , the optimal action is to serve the cached version and discard for  $\tau \leq \tau^0$ , and fetch, serve, and cache for  $\tau > \tau^0$ . We observe in Figure 2a that the area region where the optimal action to serve the cached version and discard increases as  $C_h$  increases and becomes constant after  $C_h$  exceeds  $I$ . Furthermore, the region where the optimal action is to discard, i.e.,  $\{2, 3\}$ , expands as  $C_h$  increases from 0 to  $I$ , and the region extends to infinity after  $C_h$  exceeds  $I$ .
- (b) When the state of the content is  $(\tau, 1, 0)$ , given  $C_h = 0$ , keep for regardless of the value of  $\tau$ . For  $0 < C_h < I$ , the optimal action is to keep for  $\tau \leq \bar{\tau}$  and discard for  $\tau > \bar{\tau}$ . For  $C_h \geq I$ , the optimal action is to discard for any value of  $\tau$ . We observe in Figure 2b as we increase  $C_h$ , the area of the region where the optimal action is to discard expands and extends to infinity as  $C_h$  exceeds  $I$ .
- (c) When the state of the content is  $(0, 1)$ , the optimal action is to fetch and fetch serve and discard for  $C_h < I$  and  $C_h \geq I$ , respectively.

In the next section, we establish *indexability* of the single content problem.

#### A. Indexability

In this section first we show that the single content MDP is indexable and then we compute the Whittle index for each

content  $n \in \mathcal{C}(t_k) \cup e_k$ . The passive set under the holding cost  $C_h$  is given by the states for which it is optimal to discard the content and more precisely:  $\mathcal{P}(C_h) := \{s : \pi^*(s) \in \{2, 3\}\}$ .

*Definition of Indexability:* A content is indexable if the passive set of that content satisfies the following conditions [16]:

- (a)  $\mathcal{P}(0) = \emptyset$  and  $\mathcal{P}(\infty) = \mathcal{S}$
- (b)  $\mathcal{P}(C_{h_1}) \subseteq \mathcal{P}(C_{h_2})$  for  $C_{h_1} \leq C_{h_2}$

The corresponding RMAB is indexable if the single content MDP is indexable.

*Theorem 4:* The single content MDP is indexable.

*Proof:* Recall from Table II if  $C_h = 0$ , the optimal action  $\pi^*(s) \in \{0, 1\} \forall s$ , i.e., it is optimal to keep the content for any state. Hence,  $\mathcal{P}(0) = \emptyset$ .

If  $C_h \geq I$ , then the optimal action  $\pi^*(s) \in \{2, 3\} \forall s$ , i.e., it is optimal to discard the content for any state. Hence,  $\mathcal{P}(C_h) = \mathcal{S}$  for  $C_h \geq I$  and moreover  $\mathcal{P}(\infty) = \mathcal{S}$ . Hence, for  $C_h \geq I$ ,  $\mathcal{P}(C_{h_1}) = \mathcal{P}(C_{h_2})$ . It is remaining to show that  $\mathcal{P}(C_{h_1}) \subseteq \mathcal{P}(C_{h_2})$  for  $0 < C_{h_1} \leq C_{h_2} < I$ . Recall, from Lemma 3 that  $(\tilde{\tau}_{C_h}, \bar{\tau}_{C_h})$  are the unique solutions to the following two equations:

$$\begin{aligned} 1. \quad & c_a \lambda p \beta (\tilde{\tau}_{C_h} \bar{\tau}_{C_h} - \frac{\bar{\tau}_{C_h}^2}{2}) - C_h \bar{\tau}_{C_h} + c_a \lambda \tilde{\tau}_{C_h} - c_f = 0 \\ 2. \quad & \beta (\tilde{\tau}_{C_h} - \bar{\tau}_{C_h}) + e^{-\beta(\tilde{\tau}_{C_h} - \bar{\tau}_{C_h})} - 1 - \frac{C_h}{p c_a \lambda} = 0 \end{aligned}$$

Furthermore, from Lemma 3  $\tilde{\tau}$  and  $\bar{\tau}$  increases and decreases, respectively as we increase  $C_h$ . Hence, for  $0 < C_{h_1} \leq C_{h_2} < I$ , we note that  $\bar{\tau}_{C_{h_2}} \leq \bar{\tau}_{C_{h_1}} < \tilde{\tau}_{C_{h_1}} \leq \tilde{\tau}_{C_{h_2}}$ . The passive sets corresponding to  $C_{h_1}$  and  $C_{h_2}$  are  $\mathcal{P}(C_{h_1}) = \{(\tau, 1, 1) : \bar{\tau}_{C_{h_1}} \leq \tau \leq \tilde{\tau}_{C_{h_1}}\} \cup \{(\tau', 1, 0) : \tau' \geq \bar{\tau}_{C_{h_1}}\} \cup \{(0, 0)\}$  and  $\mathcal{P}(C_{h_2}) = \{(\tau, 1, 1) : \bar{\tau}_{C_{h_2}} \leq \tau \leq \tilde{\tau}_{C_{h_2}}\} \cup \{(\tau', 1, 0) : \tau' \geq \bar{\tau}_{C_{h_2}}\} \cup \{(0, 0)\}$ , respectively. Hence,  $\mathcal{P}(C_{h_1}) \subseteq \mathcal{P}(C_{h_2})$  for  $C_{h_1} \leq C_{h_2}$ . ■

In the following we consider the Multi-content problem and obtain the Whittle indices for each content. Finally, we propose a Whittle index based policy as a solution to the problem (1) subject to (2).

#### IV. WHITTLE INDEX POLICY FOR MULTI-CONTENT PROBLEM

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

We already mentioned in Section III-A that the single content MDP is indexable and hence the RMAB is also indexable. We obtained the closed form solution of the Whittle index for the contents  $n \in \mathcal{C}(t_k) \cup \{e_k\}$ . The Content  $n \in \mathcal{C}(t_k) \cup \{e_k\}$  can be in the following states as follows:

$$s_n(t_k) = \begin{cases} (\tau_k^n, 1, 1), & \text{if } n \in \mathcal{C}(t_k) \cap \{e_k\} \\ (\tau_k^n, 1, 0) & \text{if } n \in \mathcal{C}(t_k) \setminus \{e_k\} \\ (0, 1) & \text{if } n \in \{e_k\} \setminus \mathcal{C}(t_k) \end{cases} \quad (21)$$

The Whittle index policy computes the indices for the contents in  $\mathcal{C}(t_k) \cup e_k$  and keep  $M$  contents in the cache having highest indices. As discussed in Section II one of the following scenarios may occur.

- (a) Suppose, the requested content is in the cache, i.e.,  $e_k \in \mathcal{C}(t_k)$ . We note that  $|e_k \cup \mathcal{C}(t_k)| = M$  and cache size is  $M$ ,

no content needs to be discarded. Hence, there are no need to compute the Whittle indices for each content. In this case the optimal action  $\pi^*(s_k) \in \{0, 1\}$  for Content  $e_k$  and for other contents  $\mathcal{C}(t_k) \setminus e_k$ , the optimal action is 0. Finally, from Table II, we consider the following actions for Content  $e_k$ :

- (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  $\tau^{e_k*} = -\frac{1}{p_{e_k} \beta} + \sqrt{(\frac{1}{p_{e_k} \beta})^2 + \frac{2c_f}{p_{e_k} \beta c_a \lambda_{e_k}}}$ . Since, we do not discard any content in this case and Content  $e_k$  is in state  $(\tau_k^{e_k}, 1, 1)$ , we consider the first case in Table II, where the optimal action is not to discard for state  $(\tau_k^{e_k}, 1, 1)$ .

- (b) Let us consider the case when the requested content is not in the cache, i.e.,  $e_k \notin \mathcal{C}(t_k)$ . It is then fetched at cost  $c_f$  and served. We observe that,  $|\mathcal{C}(t_k) \cup e_k| = M + 1$ . Among  $M + 1$  contents, one content needs to be discarded. The fetched content either can replace an existing content in the cache or can be discarded after serving. We calculate the Whittle indices for each content in  $\mathcal{C}(t_k) \cup e_k$  and discard the content having least Whittle index. Content  $n$  is in state  $(\tau_k^n, 1, 0)$  if  $n \in \mathcal{C}(t_k) \setminus e_k$  and  $(0, 1)$  if  $n = e_k$ . Let us denote  $W_n(s)$  as Whittle index for content  $n$  at state  $s$ . Then,

*Theorem 5:* For  $n \in \mathcal{C}(t_k) \setminus e_k$ ,

$$W_n((\tau, 1, 0)) = \begin{cases} C_h(\tau) & \text{if } \tau < \tau^{n*} \\ 0, & \text{if } \tau \geq \tau^{n*} \end{cases} \quad (22)$$

Where  $(C_h(\tau), \tilde{\tau}(\tau))$  be the unique solution to the following to equations:

$$c_a \lambda_n p_n \beta (\tilde{\tau} \tau - \frac{\tau^2}{2}) - C_h \tau + c_a \lambda_n \tilde{\tau} - c_f = 0 \quad (23)$$

$$\beta (\tilde{\tau} - \tau) + e^{-\beta(\tilde{\tau} - \tau)} - 1 - \frac{C_h}{p_n c_a \lambda_n} = 0 \quad (24)$$

and  $W_{e_k}((0, 1)) = I_{e_k} = p_{e_k} \beta c_f - p_{e_k} c_a \lambda_{e_k} (1 - e^{-\frac{\beta c_f}{c_a \lambda_{e_k}}})$ .

*Proof:* Suppose, the state of Content  $n$  is  $(\tau, 1, 0)$ . Then the Whittle index will be minimum  $C_h$  for which  $\pi^*(\tau, 1, 0) = 2$ . From Table II, we note that the optimal action is 2 for some  $C_h \in (0, I_n]$ , where  $I_n = p_n \beta c_f - p_n c_a \lambda_n (1 - e^{-\frac{\beta c_f}{c_a \lambda_n}})$ . From Theorem 2, the Whittle index will be  $\min\{C_h : C_h \text{ satisfies (23) and (24)}\}$ . We note that (23) and (24) are same as (18) and (19), respectively except  $p$  being replaced by  $p_n$  and  $\lambda$  being replaced by  $\lambda_n$ . From Lemma 3, we note that there is a unique value of  $C_h$  for which (23) and (24) are satisfied. Hence, we obtain the Whittle indices for contents in  $\mathcal{C}(t_k) \setminus e_k$ . The content  $e_k$  has state  $(0, 1)$ . We note that from Table II, for  $C_h < I_{e_k}$ , the optimal action is 1 or fetch, serve and cache and for  $C_h \geq I_{e_k}$ , the optimal action is 3 or fetch, serve and discard. Hence the Whittle index for  $e_k$  is  $I_{e_k}$ . ■

We propose the Algorithm 1 based on Whittle Index as a solution to the problem (1).

---

**Algorithm 1** Whittle Index-based Caching for Freshness

---

```

1: Initialization: Start with a random cache and random
   vector  $\{\tau_k^n : n \in N\}$ 
2: procedure CACHING( )
3:   for all request epoch do
4:     if the requested content is found in the cache, i.e.,
        $e_k \in \mathcal{C}(t_k)$  then
5:       do not discard any content
6:       if  $\tau_k^{e_k}$  exceeds the threshold,  $\tau_k^{e_k*}$  then
7:         fetch and cache the requested content
8:       else
9:         serve and keep
10:      end if
11:    else
12:      fetch the requested content
13:      for all contents in the cache and requested
        content do
14:        calculate Whittle indices
15:      end for
16:      Discard the content having the least Whittle
        index
17:    end if
18:  end for
19: end procedure

```

---

#### A. Whittle Index Policy for Multi-content having Different Sizes

In this section we consider a more general problem where the contents can be of different sizes. Let  $d_n$  be the size of content  $n$ . Then the fetching cost of item  $n$  is  $c_f d_n$ , where  $c_f$  is the fetching cost of an item of unit size. We reformulate the optimal caching problem (1) as follows:

$$\begin{aligned}
& \inf_{\pi \in \Pi} \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E}_\pi \left[ \sum_{k=1}^{A(T)} \mathbb{1}_{\{a_k^{e_k}=0\}} c_a \nu_{e_k}(t_k) + \mathbb{1}_{\{a_k^{e_k} \in \{1,3\}\}} c_f d_{e_k} \right] \\
& \text{s. t. } \sum_{n=1}^N d_n \mathbb{1}_{\{a_k^n \in \{0,1\}\}} \leq M, \forall k \geq 1
\end{aligned} \quad (25)$$

Using similar computations as (3), (5) and (6) we can write the single content problem as:

$$\begin{aligned}
& \inf_{T \rightarrow \infty} \lim_{T \rightarrow \infty} \frac{1}{\mathbb{E}[A(T)]} \mathbb{E} \left[ \sum_{k=1}^{A(T)} \mathbb{1}_{\{a_k \in \{0,2\}\}} c_a \nu(t_k) + \mathbb{1}_{\{a_k \in \{1,3\}\}} d c_f \right. \\
& \quad \left. + \mathbb{1}_{\{a_k \in \{0,1\}\}} (t_{k+1} - t_k) d C_h \right]
\end{aligned} \quad (26)$$

where  $d$  replaces  $d_n$  in (25). For the single content problem also we can obtain similar result as Theorem 2 where  $C_h$  is replaced by  $d C_h$  and  $c_f$  is replaced by  $d c_f$  and prove that the single content MDP is indexable. For the multi-content problem the policy would be as follows:

(a) Suppose, the requested content is found in the cache, i.e.,  $e_k \in \mathcal{C}(t_k)$ . Then we consider the following actions for Content  $e_k$ :

- (a) 0 or serve the cached copy if  $\tau \leq \tau_1^{e_k*}$
- (b) 1 or fetch and cache if  $\tau > \tau_1^{e_k*}$

where  $\tau_1^{e_k*} = -\frac{1}{p_{e_k} \beta} + \sqrt{\left(\frac{1}{p_{e_k} \beta}\right)^2 + \frac{2c_f d_{e_k}}{p_{e_k} \beta c_a \lambda_{e_k}}}$ . We observe that  $\tau_1^{e_k*}$  has the same expression as  $\tau^{e_k*}$  in Algorithm 1 except  $c_f$  is replaced by  $c_f d_{e_k}$ .

(b) Suppose, the requested content is not found in the cache, i.e.,  $e_k \notin \mathcal{C}(t_k)$ . Then, we compute the Whittle indices of  $M+1$  contents and choose a number of top contents until the constraint (25) is violated [17].

#### V. PERFORMANCE ANALYSIS

In this section, we study the performance of the Whittle Index based policy. We consider the number of contents,  $N = 1000$ , and study the impact on average cost as we increase the cache size from 40 to 100 under the following settings [3], [18]: The popularity of the content  $n$  be  $p_n = \frac{1}{n}$ , the fetching cost and the aging cost are  $c_f = 1$  and  $c_a = 0.1$ , respectively, and the arrival rate,  $\beta = 5$ .

##### A. Comparison

The update rate of the content  $n$  at the central server is  $\lambda_n = \lambda = 0.01$ . We implement the the proposed Whittle index based policy and plot the average cost for different cache sizes. Figure 3 shows that the Whittle Index based policy outperforms the policy proposed by Abolhassani et al. [3, Theorem 2]. We numerically compute the average cost of the relaxed RMAB problem using Table II. Furthermore, the average cost under the Whittle index policy is almost same as the optimal cost of the relaxed RMAB problem. As we mentioned earlier the optimal cost of the relaxed RMAB problem acts as a lower bound of the optimal cost of the original problem, the Whittle index policy is therefore optimal.

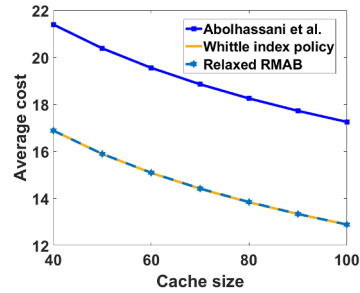


Fig. 3: Comparison of average cost between Whittle Index based policy and the policy by Abolhassani et al.[3]

##### B. Effect of update rate ( $\lambda$ ) on average cost

In this subsection, we implement the Whittle index policy to compute the average cost for three different update rates  $\lambda = 0.01, 2$  and  $5$  and for  $c_f = 5$ , while keeping all the other parameters the same as in section V. We observe that the average cost increases as the update rate increases (see Figure 4a). Since the ageing cost of a content is  $\propto$  its update rate, it is obvious the average cost will increase as we increase

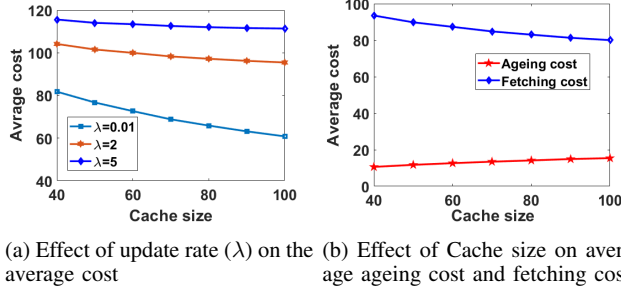


Fig. 4: Effect of update rate and cache size on average ageing cost and fetching cost

the update rate. Moreover, the average cost decreases as the cache size increases (see Figure 4a). This is because a larger cache size reduces fetching costs, given that more items can be accommodated. However, lowering fetches results in serving more items with older versions, therefore causing an increase in the aging cost. For example, see Figure 4b, where we plot the average ageing cost and fetching cost for  $\lambda = 2$ . The overall average cost decreases due to the Whittle index based policy's management of the trade-off between fetching costs and aging costs.

#### C. Effect of fetching cost ( $c_f$ ) on average cost and number of fetches during cache hits

In this subsection we will discuss how the fetching cost affects number of fetches during cache hits. We implement the Whittle index based policy for different fetching costs,  $c_f = 1, 2, 5$  and for  $\lambda = 2$ , while keeping all the other parameters the same as in section V. Since the average fetching cost increases as  $c_f$  increases, we observe that ( see Figure 5) the average cost increases as we increase ( $c_f$ ) increases.

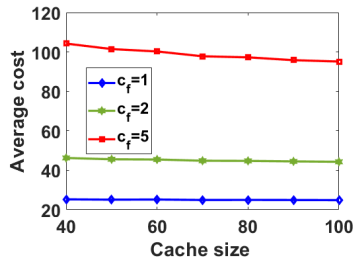


Fig. 5: Effect of fetching cost ( $c_f$ ) on average cost

## VI. CONCLUSION

We have formulated the optimal content fetching and caching problem (1) subject to hard cache capacity constraints. 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. Finally, in Theorem 5, we have computed the Whittle indices. We have proposed a Whittle index based algorithm to solve the optimal content fetching and caching problem subject to

hard cache capacity constraints. We have demonstrated that our proposed algorithm outperforms the solution provided by [3] and offers almost the same performance as the optimal policy. It would be interesting to design and solve the problem where content fetching can be unsuccessful; for example, if the local cache is connected to the central server or back-end cache via a wireless channel, the success of fetching content will depend upon the channel's reliability.

## REFERENCES

- [1] H. Zhou, L. Tang, Q. Huang, and W. Lloyd, "The evolution of advanced caching in the facebook cdn," URL: <https://research.fb.com/blog/2016/04/the-evolution-of-advanced-caching-in-the-facebook-cdn/visited-on-14/05/2021>, 2016.
- [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 *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pp. 532–543, 2001.
- [3] B. Abolhassani, J. Tadrous, A. Eryilmaz, and E. Yeh, "Fresh caching for dynamic content," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pp. 1–10, 2021.
- [4] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Information freshness and popularity in mobile caching," in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 136–140, 2017.
- [5] G. Ahani, D. Yuan, and S. Sun, "Optimal scheduling of age-centric caching: Tractability and computation," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2939–2954, 2022.
- [6] R. D. Yates, "The age of gossip in networks," in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2984–2989, IEEE, 2021.
- [7] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Achieving freshness in single/multi-user caching of dynamic content over the wireless edge," in *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, pp. 1–8, 2020.
- [8] B. Abolhassani, J. Tadrous, A. Eryilmaz, and S. Yüksel, "Optimal push and pull-based edge caching for dynamic content," *IEEE/ACM Transactions on Networking*, 2024.
- [9] B. Abolhassani, A. Eryilmaz, and T. Hou, "Swiftcache: Model-based learning for dynamic content caching in cdns," *arXiv preprint arXiv:2402.17111*, 2024.
- [10] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: evidence and implications," in *IEEE INFOCOM'99*, vol. 1, pp. 126–134, 1999.
- [11] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of applied probability*, vol. 25, no. A, pp. 287–298, 1988.
- [12] Q. Zhao, *Multi-armed bandits: Theory and applications to online learning in networks*. Springer Nature, 2022.
- [13] S. Aalto, P. Lassila, and I. Taboada, "Whittle index approach to opportunistic scheduling with partial channel information," *Performance Evaluation*, vol. 136, p. 102052, 2019.
- [14] D. P. Bertsekas et al., "Dynamic programming and optimal control 3rd edition, volume ii," Belmont, MA: Athena Scientific, vol. 1, 2007.
- [15] A. Koley and C. Singh, "Fresh caching of dynamic contents using restless multi-armed bandits," 2024.
- [16] V. Tripathi and E. Modiano, "A whittle index approach to minimizing functions of age of information," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1160–1167, 2019.
- [17] K. E. Avrachenkov and V. S. Borkar, "Whittle index policy for crawling ephemeral content," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 446–455, 2018.
- [18] B. Abolhassani, J. Tadrous, A. Eryilmaz, and E. Yeh, "Fresh caching of dynamic content over the wireless edge," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2315–2327, 2022.