# Fresh Caching for Dynamic Content

Bahman Abolhassani[1], John Tadrous[2], Atilla Eryilmaz[1], Edmund Yeh[3]

[1,2,3] Department of Electrical and Computer Engineering
[1] The Ohio State University, Columbus, 43210
[1] Email: abolhassani.2@osu.edu, eryilmaz.2@osu.edu
[2] Gonzaga University, Spokane, WA 99202
[2] Email: tadrous@gonzaga.edu
[3] Northeastern University, Boston, MA 02115
[3] Email: eyeh@ece.neu.edu

***Abstract*—We introduce a framework and provably-efficient schemes for 'fresh' caching at the (front-end) local cache of content that is subject to 'dynamic' updates at the (back-end) database. We start by formulating the hard-cache-constrained problem for this setting, which quickly becomes intractable due to the limited cache. To bypass this challenge, we first propose a flexible time-based-eviction model to derive the average system cost function that measures the system's cost due to the service of aging content in addition to the regular cache miss cost. Next, we solve the cache-unconstrained case, which reveals how the refresh dynamics and popularity of content affect the optimal caching. Then, we extend our approach to a soft-cache-constrained version, where we can guarantee that the cache use is limited with arbitrarily high probability. The corresponding solution reveals the interesting insight that 'whether to cache an item or not in the local cache?' depends primarily on its popularity level, whereas 'how long the cached item should be held in the cache before eviction?' depends primarily on its refresh rate. Moreover, we investigate the cost-cache saving tradeoffs and prove that substantial cache gains can be obtained while also asymptotically achieving the minimum cost as the database size grows.**

***Index Terms*—Content Distribution Networks, Caching, Age of Information, Dynamic Content**

## I. Introduction

The recent advances in the development of capable smart wireless devices and mobile internet services have resulted in rapidly escalating levels of data traffic over cellular networks. This surging data demand is depleting the limited spectrum resources for wireless transmission, especially over the wireless connection between the base stations and the end-users. Consequently, wireless resources are becoming scarce due to the tremendous development of throughput-hungry applications including video streaming and online gaming [1]. Thus, more sophisticated resource management strategies are needed to meet the growing demand [2].

One possible solution for tackling this problem is to cache popular contents at the users' site to reduce the total response time to data requests. Content Distribution Networks (CDNs) utilize a large mesh of caches to deliver content from locations closer to the end users [3], [4]. Existing caching strategies rely on the assumption of static (or quasi-static) nature of the stored content [5], [6]. In many real-world scenarios, such as news updates in social networks and system state updates in cyber-physical networks, the data content is subject to updates at various rates, which render the older versions of the content less useful [7]. Hence, there is a growing need to develop new caching strategies that account for the refresh characteristics and ageing costs of content for efficient dynamic-content distribution.

Broadly speaking, there are two classes of caching policies for studying the system performance: timer-based, i.e., Time-To-Live (TTL) [8], [9] and non-timer-based caching policies. In the latter case, the strongly coupled nature of the eviction policies render exact analysis difficult. In contrast, a TTL cache policy associates each content with a timer upon placement in the cache. The content is then evicted once the timer expires, independent of other cached contents. Due in part to analytical tractability [8], [10], TTL caches have been widely employed since the early days of the internet with the Domain Name System (DNS) being an important application [11]. Recently, TTL caching strategies have received renewed attention, mainly because they enable a general analytical approach which is used to model replacement-based caching policies such as Least Recently Used (LRU) [12].

Using the TTL cache refresh framework for dynamic content, [13] proposes two metrics to measure the cached content freshness: age of synchronization (AoS) and age of information (AoI). Most existing research regarding the freshness of the local cache focus on the AoI metric which was first examined in the 1990s in studies on real-time databases [14], [15].

The problem of refreshing cache contents from an AoI perspective was first formulated in [16], where a remote server generates multiple files and transmits them to a local cache. The authors assume that each file has its own request popularity, a factor that affects how often the server should update the file contained in cache. The objective is to minimize the average AoI [17]. In [18], the authors formulate the AoI problem for a system with random transmission and service processes. They show that the age decreases with increasing service rate. Nevertheless, this comes at the

This work was funded primarily by the ONR Grant N00014-19-1-2621, and in part by the NSF grants: CNS-NeTS-1514260, CNS-NeTS-1717045, CMMI-SMOR-1562065, CNS-ICN-WEN-1719371, CNS-SpecEES-1824337, CNS-NeTS-2007231, and the DTRA grant: HDTRA1-18-1-0050.

cost of increased waste in the resources spent on obsolete packets [19]. Najm et al. [20] analyze the average age and average peak of AoI under the gamma distributed service time. Sun et al. [21] study how to optimally manage the freshness of information using AoI metric and under a general age penalty function to show that a zero-wait policy does not always minimize the age. Kam et al. [22] propose a dynamic model in which the rate of requests depends on the popularity and the freshness of information to minimize the number of missed packet requests.

While AoI is a meaningful metric for measuring the freshness of content in some systems, there are many real-world scenarios where a content does not lose its value simply because time has passed since it was put into the cache. These types of dynamic contents include news and social network updates where the users prefer to have the most fresh version but so long as there is no new update, that content is considered to be the most fresh version. In this work, we use a new freshness metric called *Age-of-Version* (AoV) which counts the integer difference between the versions at the database and the local cache. We also introduce a new cost function for dynamic content caching which captures both the cost due to the miss event and the cost due to content freshness [23] which grows with the AoV metric. Moreover, our model extends the traditional caching paradigm to allow for varying *generation dynamics* of content, and calls for new designs that incorporate these dynamics into its decisions.

In particular, we propose a freshness-driven caching model for dynamic content, which accounts for the update rate of data content and provides an analysis of the average operational cost for both the constrained and unconstrained cache sizes. We aim to reveal the effect of popularity and refresh rate on the optimal caching policies. Our contributions, along with the organization of the paper, are as follows.

- In Section II, we present a tractable caching model for serving dynamic content to end users from a back-end source and formulate the general problem.
- In Section III, we attack the generally intractable problem for the special and insightful case when there is no cache constraint, i.e., all items can be stored in the cache. We characterize the optimal caching decision and explicitly identify the optimal holding time of each item in terms of its popularity and its refresh rate, which reveals the balance between the fetching cost of a fresh update and the ageing cost of serving an old version.
- In Section IV, we return to the general cost minimization problem under high-probability, and an associated average cache size constraints to propose an asymptotically optimal caching solution. The solution reveals the interesting fact that, for fresh caching of dynamic content, one should select *the items to cache based on their popularities,* while determining *the holding times of the cached items based on their refresh rates*.
- In Section V, we contrast the operational cost and average cache occupancy of the constrained cache with their counterparts in the unconstrained problem to demonstrate the potential of the proposed caching strategy and reveal the trade-off between the rate of convergence and cache size saving. The results show that the asymptotically optimal solution presented in Section IV can yield significant cache savings by discarding static items that are not sufficiently *popular*, and using the limited cache space efficiently for sufficiently popular dynamic and static items. Finally, we conclude the work in Section VI.

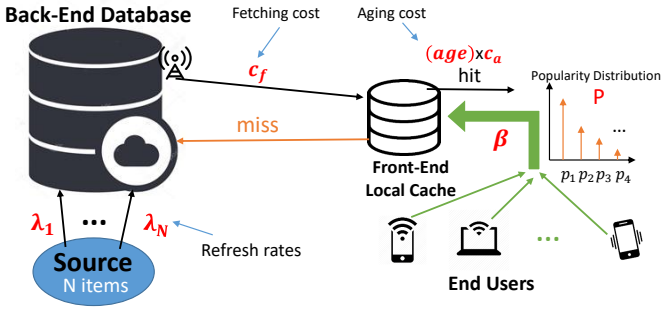


Fig. 1: Setting of *Fresh Caching for Dynamic Content*

## II. System Model

We consider the generic hierarchical setting depicted in Fig. 1, whereby: the (limited) local cache serves a user population that makes requests of content according to a popularity distribution; while the back-end database receives updates to update the content with different rates. Next, we will provide the details of this generic model, followed by the goal of our work.

**Demand Dynamics:** We assume that a set $\mathcal{N}$ of $N$ unit-size data items (with dynamically changing content) is being served to a user population by the hierarchical caching system in Fig. 1. In particular, requests arrive to the local cache according to a Poisson process[1] with rate $\beta \geq 0$, which captures the request intensity of the user population. An incoming request targets data item $n \in \mathcal{N}$ with probability $p_n$. Accordingly, the probability distribution $\mathbf{p} = (p_n)_{n=1}^N$ captures the popularity profile of the data items.

**Generation Dynamics:** At the database, each data item may receive updates to replace its previous content. We assume that data item $n$ receives updates according to a Poisson process with rate $\lambda_n \geq 0$. Note that $\lambda_n = 0$ encapsulates the traditional case of *static* content that never receives an update. We denote the vector $\boldsymbol{\lambda} = (\lambda_n)_{n=1}^N$ as the collection of update rates for the database.

**Age Dynamics:** Since the data items are subject to updates at the database, the same items in the local cache may be *older versions* of the content. To measure the freshness of local content, we define the *age* $\Delta_n(t) \in \{0, 1, \ldots\}$ at time $t$ of a cached content for item $n$ as the number of updates that the locally available item $n$ has received in the database since it has been most recently cached. We name this freshness metric as the *Age-of-Version* (AoV), since it counts the integer difference between the versions at the database and the local cache.

[1] Accordingly, we assume that the system evolves in continuous time.

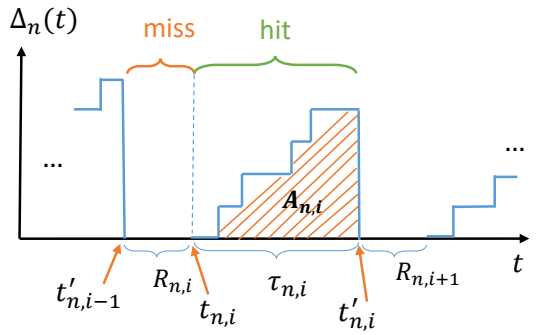


Fig. 2: Age-of-version $\Delta_n(t)$ evolution for data item $n$.

Fig. 2 illustrates an example evolution of $\Delta_n(t)$ for data item $n$ under an arbitrary holding and eviction policy. At the instant $t_{n,i}$, the local cache refreshes its content of data item $n$ for the $i^{th}$ time. This item remains in the local cache for a duration of $\tau_{n,i} \in \mathbf{R}_+$ units of time. In this sample path, the item is evicted from the local cache at time instance $t'_{n,i} = t_{n,i} + \tau_{n,i}$. During the phase $t \in [t_{n,i}, t'_{n,i})$, the AoV $\Delta_n(t)$ of item $n$ grows according to a Poisson process with rate $\lambda_n$, as governed by the aforementioned generation dynamics. At the eviction instant $t'_{n,i}$, the $\Delta_n(t)$ drops to zero by default since the next request for the item that arrives after a random duration (denoted as $R_{n,i+1}$ in the figure) will be serving a fresh update from the database.

Within the subsequent evictions $t'_{n,i}$ and $t'_{n,i+1}$ of the item $n$, we refer to: the phase $(t'_{n,i}, t'_{n,i} + R_{n,i+1}]$ as the *miss phase*, since the incoming request is not in the local cache and must be fetched from the database at a higher cost; and the phase $(t'_{n,i} + R_{n,i+1}, t'_{n,i+1}]$ as the *hit phase*, since the incoming request is served from the local cache, but possibly with a positive AoV value $\Delta_n(t)$.

**Fetching and Ageing Costs:** Now that we have the dynamics defined, we can introduce the key operational and performance costs associated with our caching system. On the operational side, we denote the cost of fetching an item from the database to the local cache by $c_f > 0$. On the performance side, we assume that serving an item $n$ from the local cache with age $\Delta_n(t)$ incurs a *freshness/age* cost of $c_a \times \Delta_n(t)$ for some $c_a \geq 0$, which grows linearly[2] with the AoV metric. This ageing cost measures the growing discontent of the user for receiving an older version of the content she/he demands.

**Problem Statement:** Our broad objective in this work is to develop efficient caching and eviction strategies for the above setting that optimally balance the tradeoff between the cost of frequently updating local content and the cost of providing aged content to the users. In particular, we are interested in provably cost-minimizing caching-and-eviction strategies that account for both the demand and the generation dynamics in order to optimally utilize a possibly limited cache space $B \in [0, \infty]$ at the local cache. We can express this goal generically as

$$\begin{aligned} &\min_{\pi \in \Pi} C^\pi \\ s.t. \quad &\sum_{n=1}^{N} X_n^\pi(t) \leq B, \quad \forall t \geq 0, \end{aligned} \tag{1}$$

where $C^\pi$ represents the mean of the combined fetching and ageing cost of the system, and $X_n^\pi(t) \in \{0, 1\}$ is the indicator that item $i$ is in the local cache under the operation of a feasible policy $\pi$. In its full generality, the feasible policy space $\Pi$ can contain any policy that decides on its fetching and eviction decisions at time $t$ with the knowledge of the cache content until time $t$ and the generation/demand dynamics[3] $(\boldsymbol{\lambda}, \beta, \mathbf{p})$, but not the ages $\{\Delta_n(t)\}_n$ (since that information depends on the updates occurring at the back-end database).

**Outline of our Approach and Results:** The generic problem in (1) falls under the scope of Partially Observable Markov Decision Processes (POMDP), and quickly becomes intractable [24]. Even formulating the problem explicitly, let alone solving it, becomes practically impossible. Therefore, a more productive approach is needed to attack this problem in order to develop algorithms and principles with performance guarantees. In this work, we propose such an approach whereby we: (i) first study the unconstrained version of the problem where $B = \infty$ in Section III, which reveals how the caching and eviction decisions must depend on the generation and demand dynamics; and then (ii) extend our approach to a constrained version in Section IV, where we can guarantee that the $B < \infty$ cache limit can be satisfied with arbitrarily high probability as the database size $N$ increases. This approach is not only productive in designing of policies with asymptotically optimal and cache-space efficient, but also reveals new and explicit *metrics* (cf. Theorems 1 and 2) for easily measuring the importance of content in terms of its popularity and refresh rates. Throughout the paper, we use cache to refer to the cache size available in the local server.

## III. Optimal Caching for Dynamic Content without Cache Constraints

In this section, we attack the generally intractable problem in (1) for the special and insightful case when there is no cache constraint, i.e., $B = \infty$. The characterization of the optimal caching decision in this section under this unconstrained setting will not only yield interesting insights about the impact of the generation dynamics, but will also form the basis of our approach to handling the cache-constrained case with high probability guarantees in Section IV.

We start by noting that the relaxation of the constraint decouples the problem into finding the optimal fetching and eviction decisions for each data item $n$ independently. This is obvious once we note that the contribution of each item to the average cost is independent of the others. This motivates

[2]While this linearity assumption is meaningful as a first-order approximation to ageing cost and facilitates simpler expressions in the analysis, it can also be generalized to convex forms to extend this basic framework.

[3]In practice, these parameters can be learned over time. Here, we assume their knowledge so that we can focus on their impact on the performance.

us in this setting to focus on a space of policies $\mathcal{T}$ with random holding times, defined next.

*Definition 1 (Policy Space $\mathcal{T}$):* $\mathcal{T}$ denotes the space of policies with random holding times, where a policy $\boldsymbol{\tau} \in \mathcal{T}$ is defined by $N$ (non-negative-valued) random variables $(\tau_n)_{n=1}^N$, representing the holding times of the items after their last fetching. In particular, the policy $\boldsymbol{\tau} = (\tau_n)_{n=1}^N$ operates as follows for each item $n \in \mathcal{N}$: (i) if item $n$ is not in the local cache when it is requested at time $t$, a fresh version of it is fetched from the database (at cost $c_f$) and served to the user; (ii) at the time of fetching item $n$ into the local cache, a random holding time is generated (independently from previous realization of holding times) with respect to the distribution of $\tau_n$, and item $n$ is held in the queue for the duration of the generated $\tau_n$ value, at which time it is evicted from the local cache; (iii) if item $n$ is in the local cache when it is requested at time $t$, it is served (with age-of-version cost of $c_a\, \Delta_n(t)$) to the user.

The space $\mathcal{T}$ takes advantage of the decoupling of the caching decisions between items as well as possesses the flexibility to adapt to different generation and demand dynamics of data items. The next lemma explicitly characterizes the average cost and average cache size of such a policy $\boldsymbol{\tau} \in \mathcal{T}$ in terms of the first and second moments of the holding time distributions of the policy $\boldsymbol{\tau}$.

*Lemma 1:* Let $C(\boldsymbol{\tau})$ and $B(\boldsymbol{\tau})$, respectively, denote the average cost and the average cache occupancy when the policy $\boldsymbol{\tau} \in \mathcal{T}$ is implemented for the caching system without cache constraints at the local cache. Then,

$$C(\boldsymbol{\tau}) = \beta \sum_{n=1}^{N} p_n \frac{\frac{1}{2} c_a \lambda_n p_n \beta\, \mathbb{E}\left[\tau_n^2\right] + c_f}{1 + \beta p_n\, \mathbb{E}\left[\tau_n\right]}, \quad (2)$$

$$B(\boldsymbol{\tau}) = \sum_{n=1}^{N} \frac{\beta p_n \mathbb{E}[\tau_n]}{1 + \beta p_n \mathbb{E}[\tau_n]}, \quad (3)$$

where $(\boldsymbol{\lambda}, \beta, \mathbf{p})$ are the system model parameters (cf. Section II) and $(\tau_n)_n$ are the random variables describing the policy $\boldsymbol{\tau}$ (c.f. Definition 1).

**Proof.** The average system cost utilizing the local cache to serve the requests comprises two main terms. Average fetching cost associated with requests that are not in the cache after a *miss* event. And, average freshness cost associated with requests that are served from the cache after a *hit* event, in which case an ageing/freshness cost is incurred due to the fact that the cached content may not be the most fresh version. Then the average cost $C(\boldsymbol{\tau})$ under the policy $\boldsymbol{\tau} \in \mathcal{T}$ can be expressed as:

$$C(\boldsymbol{\tau}) = \beta \sum_{n=1}^{N} p_n \left( (1 - h_n(\boldsymbol{\tau}))c_f + h_n(\boldsymbol{\tau}) \overline{\Delta}_n(\boldsymbol{\tau}) c_a \right), \quad (4)$$

where $\overline{\Delta}_n(\boldsymbol{\tau})$ is the time average age of the data item $n$ served from the local cache when the policy $\boldsymbol{\tau} \in \mathcal{T}$ is implemented. Based on Renewal Reward Theorem, we have:

$$\overline{\Delta}_n(\boldsymbol{\tau}) = \lim_{T \to \infty} \frac{1}{T} \int_0^T \Delta_n(t) dt = \frac{\mathbb{E}[A_{n,i}]}{\mathbb{E}[\tau_{n,i}]} = \frac{1}{2} \lambda_n \frac{\mathbb{E}[\tau_{n,i}^2]}{\mathbb{E}[\tau_{n,i}]},$$

where $A_{n,i}$ is the area shown in Fig. 2 and the last equality comes from the fact that:

$$\mathbb{E}[A_{n,i} \mid \tau_{n,i}] = \mathbb{E}[\int_0^{\tau_{n,i}} N_n(t) dt \mid \tau_{n,i}]$$

$$= \int_0^{\tau_{n,i}} \mathbb{E}[N_n(t) \mid \tau_{n,i}] dt = \int_0^{\tau_{n,i}} \lambda_n t dt = \lambda_n \frac{\tau_{n,i}^2}{2},$$

and $N_n(t)$ is a Poisson process with parameter $\lambda_n$ which is independent of $\tau_{n,i}$. Then noting that $\mathbb{E}[A_{n,i}] = \mathbb{E}[\mathbb{E}[A_{n,i} \mid \tau_{n,i}]] = \frac{\lambda_n}{2} \mathbb{E}[\tau_{n,i}^2]$ gives us the result. We omit the indices $i$ for convenience.

Next, let us denote the steady-state hit probability under the caching policy $\boldsymbol{\tau}$ as $h_n(\tau_n) = P(\bar{X}_n(\boldsymbol{\tau}) = 1)$, where $\bar{X}_n$ is the limiting distribution of $X_n(t)$ that is the indicator of whether item $n$ is in the local cache at time $t$ or not (cf. (1)). Using the illustration of Fig. 2, it is easy to confirm that the hit probability for content $n$ can be expressed as:

$$h_n(\boldsymbol{\tau}) = \frac{\mathbb{E}[\tau_n]}{\mathbb{E}[\tau_n] + \mathbb{E}[R_n]}, \quad (5)$$

where $R_n$ is the interarrival time between requests of item $n$. Since requests for item $n$ arrive at the cache according to a Poisson process with rate $\beta p_n$, we have $\mathbb{E}[R_n] = \frac{1}{\beta p_n}$. Then, substituting in (5), we get the cost expression of (2).

Using the hit probability given in (5) and noting that $\mathbb{E}[\bar{X}_n(\boldsymbol{\tau})] = h_n(\boldsymbol{\tau})$, the average cache occupancy which is $\mathbb{E}[\sum_{n=1}^N \bar{X}_n(\boldsymbol{\tau})] = \sum_{n=1}^N \mathbb{E}[\bar{X}_n(\boldsymbol{\tau})]$ gives (3). ∎

The explicit characterization of the cost under Lemma 1 allows us to pose the problem of finding the cost minimizing policy in this setting as:

$$C^*(\boldsymbol{\lambda}, \beta, \mathbf{p}) = \min_{\boldsymbol{\tau} \in \mathcal{T}} C(\boldsymbol{\tau}), \quad (6)$$

where the minimization is performed over all distributions for the holding times $(\tau_n)_n$ with non-negative ranges, and the tuple $(\boldsymbol{\lambda}, \beta, \mathbf{p})$ indicates that the solution is a function of these system parameters. For brevity, we will occasionally omit these parameters and refer to the optimal cost as $C^*$, and later on we will also use $C^*(N)$ when we study the scaling of the performance as the database size $N$ grows. The following theorem fully solves (6).

*Theorem 1:* Policy $\boldsymbol{\tau}^* \in \mathcal{T}$ that solves (6) is given by:

$$\tau_n^* = \begin{cases} \frac{1}{\beta p_n} \left( \sqrt{1 + 2 \frac{\beta p_n c_f}{c_a \lambda_n}} - 1 \right), & n \in \mathcal{D}, \\ \infty, & n \in \mathcal{S}, \end{cases} \quad (7)$$

where $\mathcal{D} = \{n \in \mathcal{N} \mid \lambda_n > 0\}$, and $\mathcal{S} = \{n \in \mathcal{N} \mid \lambda_n = 0\} = \mathcal{N} \backslash \mathcal{D}$ are, respectively, the set of *dynamic* and *static* data items. Then, the corresponding optimal average cost is given by:

$$C^*(\boldsymbol{\lambda}, \beta, \mathbf{p}) = \sum_{n \in \mathcal{D}} c_a \lambda_n \left( \sqrt{1 + 2 \frac{\beta p_n}{\lambda_n} \frac{c_f}{c_a}} - 1 \right). \quad (8)$$

Also, the average cache occupancy under $\boldsymbol{\tau}^*$ is given by:

$$B(\boldsymbol{\tau}^*) = |\mathcal{S}| + \sum_{n \in \mathcal{D}} \frac{\beta p_n \tau_n^*}{1 + \beta p_n \tau_n^*}. \quad (9)$$

**Proof.** First we show that the average system cost given in (2) is minimized when the variable $\tau_n$ is a constant, $\forall n$. For a random variable $\tau_n$ with expectation $\mathbb{E}[\tau_n]$, in order

to minimize the cost, the second moment $\mathbb{E}[\tau_n^2]$ should be minimum. Since $var[\tau_n] = \mathbb{E}[\tau_n^2] - (\mathbb{E}[\tau_n])^2 \geq 0$, so the minimum possible is $\mathbb{E}[\tau_n^2] = (\mathbb{E}[\tau_n])^2$ which is a constant random variable. In calculating (2) we assumed that steady state distribution exists for the given random variable $\tau_n$. Now we verify it for the constant random variable $\tau_n$.

*Lemma 2:* For a constant random variable $\tau_n$, Bernoulli process $(X_n(t), t \geq 0)$ has a steady state distribution and its average is given by (5).

**Proof.** According to Fig. 2, for data item $n$, define a new random variable $Z_n = R_n + \tau_n$ where $\tau_n$ is constant. We have $\mathbb{E}[Z_n] = \tau_n + \frac{1}{\beta p_n}$. Based on the instances of random variable $Z_n$, define $S_n^q = \sum_{i=1}^{q} Z_{n,i} = \sum_{i=1}^{q}(R_{n,i} + \tau_n)$ to be the time of the $q^{th}$ renewal when the data item $n$ enters the cache for the $q^{th}$ time. Let $W(t)$ be the number of times that data item $n$ has evicted from the cache up to time $t$. Blackwell's renewal theorem states that for any fixed $\tau_n > 0$:

$$\lim_{t\to\infty} [\mathbb{E}[W(t+\tau_n)] - \mathbb{E}[W(t)]] = \frac{\tau_n}{\mathbb{E}[Z_n]} = \frac{\beta p_n \tau_n}{1+\beta p_n \tau_n},$$

which shows the existence of the steady state distribution. Therefore the random process $(X_n(t), t)$ has a steady state distribution with its average given by the above equation.

■

The cost minimization problem for the unconstrained cache can thus be expressed as:

$$C^*(\boldsymbol{\lambda}, \beta, \mathbf{p}) = \min_{\tau_n \geq 0} \beta \sum_{n=1}^{N} p_n \frac{\frac{1}{2} c_a \lambda_n p_n \beta \tau_n^2 + c_f}{1+\beta p_n \tau_n}.$$

The objective function has the form of *quadratic over linear* ratio, which is convex. Using KKT conditions gives the optimal solution for $\boldsymbol{\tau}^*$ in (7). Substituting $\boldsymbol{\tau}^*$ in (2) gives the optimal cost of (8).

To prove the optimal average cache occupancy (9), substituting the optimal solution (7) in the definition of average cache occupancy given in Lemma 1 and noting that $\tau_n^* = \infty, \forall n \in \mathcal{S}$, we obtain $\sum_{n\in\mathcal{S}} \frac{\beta p_n \tau_n^*}{1+\beta p_n \tau_n^*} = |\mathcal{S}|$ which completes the proof. ■

Theorem 1, under the unconstrained cache setting, provides some useful insights about the nature of the optimal caching strategy for dynamic content: (i) we see that the cost minimizing policy $\boldsymbol{\tau}^*$ selects a fixed holding time for each item $n$ rather than any other random choice; (ii) more interestingly, (7) explicitly characterizes the optimal holding time of each dynamic item $n$ in terms of its popularity $p_n$ and its refresh rate $\lambda_n$ in order to strike the optimal balance between the fetching cost of a fresh update and the ageing cost of serving an old version; (iii) less interestingly, we also see that any static item is cached forever under this unconstrained setting since it is never necessary to update it once it is fetched; and (iv) it explicitly characterizes the average cache occupancy of $\tau^*$ in terms of system parameters.

In the next section, we will build upon this foundation to return to a soft-constrained version of the problem (1).

## IV. Asymptotically-Optimal Caching for Dynamic Content with cache Constraints

Returning the general cost minimization problem given in (1), the instantaneous cache size constraint with $B < \infty$ entails a dependence between the optimizing items holding time. With such a dependence the optimization (1) suffers from the curse of dimensionality and has no tractable solution. In this section, we bypass this challenge by replacing the deterministic-constraint $\sum_{n=1}^{N} X_n^{\pi}(t) \leq B$, at all times $t$, to a probabilistic-constraint where cache size limit has to be met with (arbitrarily) high probability over time. In particular let us introduce the following probabilistic version of (6):

$$\begin{aligned} \min_{\boldsymbol{\tau}\in\mathcal{T}} \quad & C(\boldsymbol{\tau}) \\ s.t. \quad & P\left(\sum_{n=1}^{N} \bar{X}_n(\boldsymbol{\tau}) \leq B\right) \geq 1-\delta, \end{aligned} \tag{10}$$

for any arbitrarily small $\delta > 0$, where $\bar{X}_n(\boldsymbol{\tau})$ is the steady-state fraction of time that item $n$ is held in the cache under policy $\boldsymbol{\tau}$. Such probabilistic approaches to solving deterministic problems are used increasingly frequently and fruitfully in learning and optimization domains. Solving this high-probability variation of the hard problem, in turn, provides a means to operate the original system efficiently with arbitrarily high probability.

Despite its softer statistical form, solving (10) is still complicated by the need to design with guarantees in the tail distribution of its cache use. To tackle this challenge, we, in turn, pose the following average-cache-constrained problem with a flexible choice of cache size bound $\tilde{B} \in [0, \infty)$ :

$$\begin{aligned} \min_{\boldsymbol{\tau}\in\mathcal{T}} \quad & C(\boldsymbol{\tau}) \\ s.t. \quad & B(\boldsymbol{\tau}) \leq \tilde{B}, \end{aligned} \tag{11}$$

where $B(\boldsymbol{\tau})$ is the average cache occupancy under the policy $\boldsymbol{\tau}$ that is explicitly characterized in (3). We note that this problem is non-convex since the constraint set $\{\boldsymbol{\tau} : B(\boldsymbol{\tau}) \leq \tilde{B}\}$ is non-convex. Nevertheless, the approach in the rest of the section is to first solve the non-convex problem (11) for any given $\tilde{B}$, and then choose a particular $\tilde{B}$ as a function of the given $B < \infty$ and $\delta > 0$ in order to guarantee the probabilistic constraint in (10). Accordingly, we first provide the solution of (11) in the next theorem.

*Theorem 2:* Policy $\tilde{\boldsymbol{\tau}}^* = (\tilde{\tau}_n^*)_n \in \mathcal{T}$ that solves (11) is given by deterministic $\tilde{\tau}_n^* \geq 0$, $\forall n$, and $\tilde{\alpha}^* \geq 0$ satisfying:

$$\tilde{\tau}_n^* = \left\{ \begin{array}{ll} \frac{1}{\beta p_n}\left[\sqrt{1+2\frac{\beta p_n c_f - \tilde{\alpha}^*}{c_a \lambda_n}} - 1\right]^+, & \forall n \in \mathcal{D} \\ \infty, & \forall n \in \mathcal{S}, \quad \tilde{\alpha}^* < \beta p_n c_f \\ \in [0,\infty], & \forall n \in \mathcal{S}, \quad \tilde{\alpha}^* = \beta p_n c_f \\ 0, & \forall n \in \mathcal{S}, \quad \tilde{\alpha}^* > \beta p_n c_f \end{array} \right\}, \tag{12}$$

where $[z]^+ = \max(0, z)$, and

$$\tilde{\alpha}^*\left(B(\tilde{\boldsymbol{\tau}}^*) - \tilde{B}\right) = 0, \qquad B(\tilde{\boldsymbol{\tau}}^*) \leq \tilde{B}, \tag{13}$$

where $\mathcal{D}$ and $\mathcal{S}$ are, respectively, the set of *dynamic* and *static* data items defined in Theorem 1.

**Proof.** In the proof of Theorem 1, we showed that in order to minimize the cost, the random variable $\tau_n$ should be a constant. Also, Lemma 2 shows that for such a constant random variable $\tau_n$, the Bernoulli process $(X_n(t))_t$ has a steady-state distribution whose average is given by (5). Therefore the assumptions to calculate the average cost and average cache occupancy given in (2) and (3) hold and the optimization problem (11) can be rewritten as:

$$\min_{\tau_n \geq 0} \beta \sum_{n=1}^{N} p_n \frac{\frac{1}{2} c_a \lambda_n p_n \beta \tau_n^2 + c_f}{1 + \beta p_n \tau_n}$$
$$s.t. \quad \sum_{n=1}^{N} \frac{\beta p_n \tau_n}{1 + \beta p_n \tau_n} \leq \tilde{B}.$$

This is not a convex optimization problem. However, we take the following approach to solve it. Define the feasible set $\mathcal{F}_B$ as:

$$\mathcal{F}_B = \left\{ (\tau_1, \cdots, \tau_N) \,|\tau_n \geq 0, g(\tau) = \sum_{n=1}^{N} \frac{\beta p_n \tau_n}{\beta p_n \tau_n + 1} \leq \tilde{B} \right\}$$

which is a non-convex set. Then the cost optimization problem (11) can be expressed as:

$$\min_{\boldsymbol{\tau} \in \mathcal{F}_B} C(\boldsymbol{\tau}). \tag{14}$$

For any optimization problem $\min_{\boldsymbol{\tau} \in \mathcal{F}} C(\boldsymbol{\tau})$ as it is given in [25], if all the following hold:

1) Slater condition,

2) non degeneracy assumption for $\forall \boldsymbol{\tau} \in \mathcal{F}$,

3) $\exists \boldsymbol{\tau}' \in \mathcal{F} : \forall \boldsymbol{\tau} \in \mathcal{F}, \exists \mathbf{t}_n \downarrow 0$ with $\boldsymbol{\tau}' + \mathbf{t_n}(\boldsymbol{\tau} - \boldsymbol{\tau}') \in \mathcal{F}$,

4) $L_C(\boldsymbol{\tau}) = \left\{ \boldsymbol{\tau}' \in R^N : C(\boldsymbol{\tau}') < \mathrm{C}(\boldsymbol{\tau}) \right\}$ is a convex set,

then if $\boldsymbol{\tau}$ is a non trivial KKT point, it is a global minimizer.

*Lemma 3:* Optimization problem (14) satisfies all the above four necessary conditions.

**Proof.** (Lemma 3) Please refer to Appendix A. ■

Therefore, the non-trivial KKT solution to the problem (14) would be a global minimizer. Such a solution can be expressed as:

$$\tilde{\tau}_n^* = \frac{1}{\beta p_n} \left[ \sqrt{1 + \frac{\beta p_n c_f + \frac{\tilde{\mu}_n^*}{\beta p_n} - \tilde{\alpha}^*}{\frac{c_a \lambda_n}{2} - \frac{\tilde{\mu}_n^*}{\beta p_n}}} - 1 \right] \geq 0,$$

where $\tilde{\alpha}^* \geq 0$ and $\tilde{\mu}_n^* \geq 0$ are the optimal Lagrange multipliers which satisfy all the following KKT conditions:

$$\tilde{\mu}_n^* \tilde{\tau}_n^* = 0, \quad \sum_{n=1}^{N} \frac{\beta p_n \tilde{\tau}_n^*}{1 + \beta p_n \tilde{\tau}_n^*} \leq \tilde{B},$$
$$\tilde{\alpha}^* \left( \sum_{n=1}^{N} \frac{\beta p_n \tilde{\tau}_n^*}{\beta p_n \tilde{\tau}_n^* + 1} - \tilde{B} \right) = 0.$$

Accordingly, for dynamic data items, $n \in \mathcal{D}$, with $\lambda_n > 0$, we have:

$$\tilde{\tau}_n^* = \max \left( 0, \frac{1}{\beta p_n} [\sqrt{1 + 2 \frac{\beta p_n c_f - \tilde{\alpha}^*}{c_a \lambda_n}} - 1] \right),$$

while for static data items, $n \in \mathcal{S}$, with $\lambda = 0$, we have:

$$\tilde{\tau}_n^* = \begin{cases} \infty & \tilde{\alpha}^* < \beta p_n c_f, \\ \in [0, \infty] & \tilde{\alpha}^* = \beta p_n c_f, \\ 0 & \tilde{\alpha}^* > \beta p_n c_f, \end{cases}$$

with $\tilde{\alpha}^* \geq 0$ chosen such that $\tilde{\alpha}^* \left( \sum_{n=1}^{N} \frac{\beta p_n \tilde{\tau}_n^*}{\beta p_n \tilde{\tau}_n^* + 1} - \tilde{B} \right) = 0$ and $\sum_{n=1}^{N} \frac{\beta p_n \tilde{\tau}_n^*}{\beta p_n \tilde{\tau}_n^* + 1} \leq \tilde{B}$. This completes the proof. ■

The form of the optimal solution in (12) reveals the interesting insights that, for dynamic content $n \in \mathcal{D}$: whether to cache an item depends on whether it is sufficiently popular (in particular, whether $p_n \lessgtr \frac{\tilde{\alpha}^*}{\beta c_f}$); and how long a cached item will remain in the cache before eviction depends on its refresh rate $\lambda_n$ as characterized in (12). It can also be seen that, for the same system parameters $(\boldsymbol{\lambda}, \beta, \mathbf{p})$, as the average cache limit $\tilde{B}$ decreases, then the optimal $\tilde{\alpha}^*$ that solves (12)-(13) will increase. Then, for both static and dynamic content, the popularity threshold $\tilde{\alpha}^*/(\beta c_f)$ for caching or not caching the content increases to make sure only sufficiently popular items are cached.

Now that we solved the average-cache-constrained problem (11), we are ready to connect it to the probabilistic problem (10) with the following proposition.

*Proposition 1:* For any finite $B > 0$ and arbitrarily small $\delta > 0$, there exits $\tilde{B}(\delta) = Be^{-v}$ with

$$v = \min \left\{ v' \in \mathbb{N} | \exp\left( -B \left( (v' - 1) + e^{-v'} \right) \right) \leq \delta \right\},$$

such that the solution $\tilde{\boldsymbol{\tau}}^*(\delta)$ of (11) for $\tilde{B} = \tilde{B}(\delta)$ satisfies

$$P\left( \sum_{n=1}^{N} \bar{X}_n(\tilde{\boldsymbol{\tau}}^*(\delta)) \leq B \right) \geq 1 - \delta.$$

**Proof.** (Proposition 1) Notice that $\bar{X}_n(\boldsymbol{\tau}), \forall n \in \mathcal{N}$ are independent Bernoulli random variables. We define a new random variable $Y_N(\boldsymbol{\tau}) = \sum_{n=1}^{N} \bar{X}_n(\boldsymbol{\tau})$, which is the sum of $N$ independent Bernoulli random variables and is known to have a Poisson Binomial distribution. Also using the linear property of expectation and given that $\mathbb{E}[\bar{X}_n(\boldsymbol{\tau})] = h_n(\boldsymbol{\tau})$, we have:

$$\mathbb{E}[Y_N(\boldsymbol{\tau})] = \sum_{n=1}^{N} \mathbb{E}[\bar{X}_n(\boldsymbol{\tau})] = \sum_{n=1}^{N} \frac{\beta p_n \mathbb{E}[\tau_n]}{1 + \beta p_n \mathbb{E}[\tau_n]}.$$

For the random variable $Y_N$ with Poisson Binomial distribution, using the Chernoff bound we have:

$$P(Y_N \geq B) \leq \exp(-B \log B + B + B \log(\mathbb{E}[Y_N]) - \mathbb{E}[Y_N]).$$

Then to guarantee $P(Y_N \leq B) \geq 1 - \delta$, we have:

$$-B \log B + B + B \log(\mathbb{E}[Y_N]) - \mathbb{E}[Y_N] \leq \log(\delta).$$

In this equation, setting $\delta$ to the form $\delta = \exp(-((v-1)e^v + 1)\mathbb{E}[Y_N]), \forall v \geq 1$, will give us the range of possible $\mathbb{E}[Y_N]$ as $\mathbb{E}[Y_N] \leq Be^{-m}$ to ensure that $P(Y_N \leq B) \geq 1 - \delta$ holds. Hence the choice $\tilde{B}(\delta) = Be^{-v}$. ■

Proposition 1 provides an explicit means of using the tractable problem (11) to find efficient feasible solutions to the problem (10). To glean an insight on the structure of $\tilde{B}(\delta)$, suppose that $m = 1$ and $\delta = e^{-B/e}$, which is very small for sufficiently large $B$. Then, we have $\tilde{B}(\delta) = Be^{-1}$.

In the next section, we will study the cost and cache occupancy performance merits of the proposed approximate optimization of (10) for large databases, which is commonly the case in content distribution networks. In particular, we will introduce the variable $0 \leq m(N) \leq N$ as the number of most popular items that will remain in the cache after being fetched for the optimized holding times from (12). The remaining $N - m(N)$ items will never be cached, i.e., will only be fetched and served upon a user request and not

cached. Then we will examine the cost-cache trade-off for this proposed strategy to show its desirable characteristics.

## V. Cost and cache Space Performance Analysis

To establish the performance merits or the proposed approximate solution $(\tilde{\boldsymbol{\tau}}^*, \tilde{\alpha}^*)$ given in (12) and (13), we contrast the operational cost and average cache occupancy of the approximate problem (11) with its counterpart of the unconstrained problem (6) in the asymptotic regime as the number of data items, $N$, grows.

We expose the dependence of the relevant quantities on $N$ to highlight its impact on the analysis as follows. We denote the optimal cost and average cache occupancy of (6), respectively, by $C^*(N)$ and $B^*(N)$, whereas the cost and average cache occupancy of the proposed approximate problem (11) are denoted by $\tilde{C}^{\tilde{\alpha}}(N)$ and $\tilde{B}^{\tilde{\alpha}}(N)$, where the superscript $\tilde{\alpha}$ indicates the dependence of these values to the $\tilde{\alpha}$ parameter that is optimized in (12) and (13) for a given cache bound. Here, $\tilde{\alpha} \geq 0$ is a flexible parameter that allows us to explore the tradeoff between the cost and the cache occupancy. Note that $C^*(N) = C^*(\boldsymbol{\lambda}, \beta, \mathbf{p})$ in (8) and $B^*(N) = B(\boldsymbol{\tau}^*)$ in (9). In addition,

$$\tilde{C}^{\tilde{\alpha}}(N) = \beta \sum_{n \in \mathcal{N}} p_n \frac{\frac{1}{2} c_a \lambda_n p_n \beta (\tilde{\tau}_n^{\tilde{\alpha}})^2 + c_f}{1 + \beta p_n \tilde{\tau}_n^{\tilde{\alpha}}},$$

where $(\tilde{\boldsymbol{\tau}}^{\tilde{\alpha}}, \tilde{\alpha})$ satisfies (12) and (13) for a given $\tilde{\alpha} \geq 0$ with the appropriate choice of $\tilde{B}^{\tilde{\alpha}}$ as the corresponding cache limit in (10).

As the number of data items $N$ grows, both the set of static items, $\mathcal{S}$, and/or the set of dynamic items, $\mathcal{D}$, grow in size accordingly, yet at different rates with $N$. Nevertheless, by the definition of $\mathcal{D}$ in Theorem 1, we can guarantee a minimum content update rate $\lambda_{min} > 0$ for all the items $n \in \mathcal{D}$ for any number of data items $N$. That is, $\lambda_{min} = \inf_{n \in \mathcal{D}} \lambda_n > 0, \forall n \in N$.

Further, for any given $\tilde{\alpha} \geq 0$, we define the set of popular items in the approximate problem (11), $\mathcal{P}^{\tilde{\alpha}}$, as

$$\mathcal{P}^{\tilde{\alpha}} = \left\{ n \in \mathcal{N} \mid p_n > \frac{\tilde{\alpha}}{\beta c_f} \right\}, \tag{15}$$

to contain all the items that should be held in the cache after being fetched from the back-end database since (12) implies:

$$\tilde{\tau}_n^{\tilde{\alpha}} \begin{cases} > 0, & n \in \mathcal{P}^{\tilde{\alpha}}, \\ = 0, & n \in \mathcal{N} - \mathcal{P}^{\tilde{\alpha}}. \end{cases} \tag{16}$$

It is worth noting that, if $\tilde{\alpha} = 0$, then $\mathcal{P}^{\tilde{\alpha}} = \mathcal{N}$ and all data items are considered popular which collapses to the case of the unconstrained cached size optimization (6). The last step before stating the asymptotic gains of the proposed policy is to divide the set of static items into two disjoint subsets. A subset $\mathcal{S}^{\tilde{\alpha}}$ of static items that are *popular*, i.e., $\mathcal{S}^{\tilde{\alpha}} = \mathcal{S} \cap \mathcal{P}^{\tilde{\alpha}}$, and a subset $\overline{\mathcal{S}}^{\tilde{\alpha}} = \mathcal{S} - \mathcal{S}^{\tilde{\alpha}}$ of static *unpopular* items.

The following theorem jointly establishes the asymptotic optimality of the proposed approximate policy together with characterizing the cost-cache size trade off.

*Theorem 3:* For a given $\tilde{\alpha} \geq 0$, consider the policy $\tilde{\boldsymbol{\tau}}^{\tilde{\alpha}}$ that solves (10) for a corresponding average cache bound $\tilde{B}^{\tilde{\alpha}}$ and average cost $\tilde{C}^{\tilde{\alpha}}$. Let $m(N) = |\mathcal{P}^{\tilde{\alpha}}|$ denote the number of sufficiently popular items that will be cached under $\tilde{\boldsymbol{\tau}}^{\tilde{\alpha}}$ policy.

(i) (Asymptotic Optimality) If

$$m(N) = \min(\omega(\sqrt{N}), \omega(|\overline{\mathcal{S}}^{\tilde{\alpha}}|)),$$

then:

$$\lim_{N \to \infty} \tilde{C}^{\tilde{\alpha}}(N) - C^*(N) = 0.$$

(ii) (Cost-cache Size Trade off) If $m(N) = N^b, 0 < b < 1$ and $|\overline{\mathcal{S}}^{\tilde{\alpha}}| = N^{a_2}, 0 \leq a_2 < 1$, with $b > \min(\frac{1}{2}, a_2)$, the rate of convergence is at least:

$$\tilde{C}^{\tilde{\alpha}}(N) - C^*(N) \leq O\left(N^{-\min(b-a_2, 2b-1)}\right),$$

the average cache saving is lower bounded by:

$$B^*(N) - \tilde{B}^{\tilde{\alpha}}(N) \geq |\overline{\mathcal{S}}^{\tilde{\alpha}}| = N^{a_2},$$

and the average cache occupancy $\tilde{B}^{\alpha}(N)$ is bounded by:

$$\tilde{B}^{\tilde{\alpha}}(N) \leq |\mathcal{S}^{\tilde{\alpha}}| + \frac{\beta c_f}{c_a \lambda_{min}},$$

**Proof.** Without loss of generality, assume that $p_1 \geq p_2 \geq \cdots \geq p_N > 0$. Since $m(N) = |\mathcal{P}^{\tilde{\alpha}}|$ and according to the definition of the set of popular items $\mathcal{P}^{\tilde{\alpha}}$ given in (15), we will have $\tilde{\alpha} \leq \beta c_f p_{m(N)}$ for any given $\tilde{\alpha}$ where $p_{m(N)}$ is the probability of the $m(N)^{th}$ most popular item. Using the expressions for $\tau_n^*$ and $\tilde{\tau}_n^{\tilde{\alpha}}$ given in (7) and (12) respectively, for dynamic data items we can show that:

$$\tau_n^* - \tilde{\tau}_n^{\tilde{\alpha}} \leq \frac{c_f}{c_a \lambda_n} \frac{p_{m(N)}}{p_n} \frac{1}{1 + \beta p_n \tilde{\tau}_n^{\tilde{\alpha}}}, \quad \forall n \in \mathcal{D}.$$

Since $\tau_n^* \geq \tilde{\tau}_n^{\tilde{\alpha}}, \forall n \in \mathcal{N}$, applying Taylor series to average cost of the data item $n$ will give us the following inequality:

$$\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^* \leq -\nabla \tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}})(\tau_n^* - \tilde{\tau}_n^{\tilde{\alpha}}), \quad \forall n \in \mathcal{D}. \tag{17}$$

The Lagrangian function $L(\tilde{\boldsymbol{\tau}}_n^{\tilde{\alpha}}, \tilde{\alpha}, \tilde{\boldsymbol{\mu}})$ of (10) takes the form:

$$\sum_{n=1}^{N} \tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) + \tilde{\alpha}\left(\sum_{n=1}^{N} \frac{\beta p_n \tilde{\tau}_n^{\tilde{\alpha}}}{\beta p_n \tilde{\tau}_n^{\tilde{\alpha}} + 1} - \tilde{B}(\tilde{\tau}_n^{\tilde{\alpha}})\right) + \sum_{n=m(N)}^{N} \tilde{\mu}_n \tilde{\tau}_n^{\tilde{\alpha}},$$

where $\tilde{\alpha} \geq 0$ and $\tilde{\mu}_n \geq 0, \forall n \in \{1, 2, ..., N\}$ are Lagrange multipliers. Note that since $\tilde{\tau}_n^{\tilde{\alpha}} > 0, \forall n \leq m(N)$, we have that $\tilde{\mu}_n = 0, \forall n \leq m(N)$. Using the fact that $\tilde{\boldsymbol{\tau}}_n^{\tilde{\alpha}}$ is a non-trivial KKT point for a given $\tilde{\alpha} \leq \beta c_f p_{m(N)}$ and setting the derivative of Lagrangian function to zero, we have:

$$-\nabla \tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) \leq \frac{\beta^2 p_n p_{m(N)} c_f}{(1 + \beta p_n \tilde{\tau}_n^{\tilde{\alpha}}))^2}, \quad \forall n \in \mathcal{P}^{\tilde{\alpha}}$$

$$\begin{aligned} -\nabla \tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) \leq & \beta^2 p_n p_{m(N)} c_f \\ & + \beta^2 c_f p_n (p_{m(N)} - p_n), \forall n \in \mathcal{N} - \mathcal{P}^{\tilde{\alpha}} \end{aligned}$$

Apply (17) to each popular dynamic data item $n \in \mathcal{D} \cap \mathcal{P}^{\tilde{\alpha}}$:

$$\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^* \leq \frac{\beta^2 c_f^2}{c_a} \frac{p_{m(N)}^2}{\lambda_n} \frac{1}{(1 + \beta p_n \tilde{\tau}_n^{\tilde{\alpha}})^3} \leq \frac{\beta^2 c_f^2 p_{m(N)}^2}{c_a \lambda_n},$$

and apply it to each unpopular dynamic item $n \in \mathcal{D} - \mathcal{P}^{\tilde{\alpha}}$:

$$\begin{aligned} \tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^* &\leq \frac{\beta^2 c_f^2}{c_a} \frac{p_{m(N)}^2 + p_n(p_{m(N)} - p_n)}{\lambda_n} \\ &\leq \frac{\beta^2 c_f^2}{c_a} \frac{1}{\lambda_n} \frac{5}{4} p_{m(N)}^2, \end{aligned}$$

where the second inequality comes from the fact that $p_n(p_{m(N)} - p_n) \leq \frac{1}{4}p_{m(N)}^2$.

For popular static items $n \in \mathcal{S}^{\tilde{\alpha}}$, we have $\tau_n^* = \tilde{\tau}_n^{\tilde{\alpha}} = \infty$ according to (7) and (12) respectively. Therefore, we have $\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) = C_n^* = 0, \forall n \in \mathcal{S} \cap \mathcal{P}^{\tilde{\alpha}}$. For unpopular static items $n \in \overline{\mathcal{S}}^{\tilde{\alpha}}$, $\tau_n^* = \infty$ according to (7) and according to (16) we have $\tilde{\tau}_n^{\tilde{\alpha}} = 0$. This gives $C_n^* = 0$ and $\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) = \beta p_n c_f$ based on the average cost function given in (2). Therefore,

$$\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^* = \beta p_n c_f, \quad \forall n \in \overline{\mathcal{S}}^{\tilde{\alpha}}.$$

Thus, the total average system cost is upper-bounded as:

$$\begin{aligned}
&\sum_{n=1}^{N}[\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^*] = \sum_{n\in\mathcal{D}\cap\mathcal{P}^{\tilde{\alpha}}}[\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^*] \\
&+ \sum_{n\in\mathcal{D}-\mathcal{P}^{\tilde{\alpha}}}[\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^*] + \sum_{n\in\mathcal{S}^{\tilde{\alpha}}}[\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^*] \\
&+ \sum_{n\in\overline{\mathcal{S}}^{\tilde{\alpha}}}[\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^*] \\
&\leq \frac{\beta^2 c_f^2}{c_a}\frac{5}{4}p_{m(N)}^2 \sum_{n\in\mathcal{D}}\frac{1}{\lambda_n} + \beta c_f \sum_{n\in\mathcal{S}-\mathcal{P}^{\tilde{\alpha}}} p_n
\end{aligned}$$

Since $|\mathcal{D}| = N - |\mathcal{S}|$, then $\sum_{n\in\mathcal{D}}\frac{1}{\lambda_n} \leq \frac{N-|\mathcal{S}|}{\lambda_{min}}$. Also, for unpopular static items we have $p_n \leq p_{m(N)}, \forall n \in \overline{\mathcal{S}}^{\tilde{\alpha}}$. Therefore we have that $\sum_{n\in\overline{\mathcal{S}}^{\tilde{\alpha}}} p_n \leq p_{m(N)}|\overline{\mathcal{S}}^{\tilde{\alpha}}|$. Finally, since we assumed that items are ordered based on their popularity and $p_{m(N)}$ is the probability of $m(N)^{th}$ most popular item, so $p_{m(N)} \leq \frac{1}{m(N)}$. This gives us:

$$\sum_{n=1}^{N}\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^* \leq \frac{5}{4}\frac{\beta^2 c_f^2}{c_a\lambda_{min}}\frac{N-|\mathcal{S}|}{m^2(N)} + \beta c_f\frac{|\overline{\mathcal{S}}^{\tilde{\alpha}}|}{m(N)}.$$

In order to make sure that the upper bound vanishes as $N$ increases, we need[4] to have $m(N) = \min(\omega(\sqrt{N}), \omega(|\overline{\mathcal{S}}^{\tilde{\alpha}}|))$. This proves (i). To prove (ii), note that $m(N) = \min(\omega(\sqrt{N}), \omega(|\overline{\mathcal{S}}^{\tilde{\alpha}}|))$ is equivalent to $b > \min(\frac{1}{2}, a_2)$. Then the convergence rate of the upper bound becomes:

$$\begin{aligned}
\sum_{n=1}^{N}\tilde{C}_n(\tilde{\tau}_n^{\tilde{\alpha}}) - C_n^* &= \tilde{C}^{\tilde{\alpha}}(N) - C^*(N) \\
&= O\left(N^{-\min(b-a_2, 2b-1)}\right)
\end{aligned}$$

which demonstrates the smallest rate of convergence on the average cost. On the other hand, since $m(N)$ is the number of most popular items that we choose to cache while discarding all the other unpopular ones, we can show that:

$$\tilde{B}^{\tilde{\alpha}}(N) = \sum_{n=1}^{m(N)}\frac{\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}}{1+\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}} = |\mathcal{S}^{\tilde{\alpha}}| + \sum_{n\in\mathcal{P}^{\tilde{\alpha}}-\mathcal{S}}\frac{\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}}{1+\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}},$$

where $\tilde{B}^{\tilde{\alpha}}(N)$ is the average cache occupancy under the proposed strategy. On the other hand, for the unconstrained cache system, we have:

$$B^*(N) = \sum_{n=1}^{N}\frac{\beta p_n\tau_n^*}{1+\beta p_n\tau_n^*} = |\mathcal{S}^{\tilde{\alpha}}| + |\overline{\mathcal{S}}^{\tilde{\alpha}}| + \sum_{n\in\mathcal{D}}\frac{\beta p_n\tau_n^*}{1+\beta p_n\tau_n^*}.$$

[4] $f(n) = \omega(g(n))$ means that for any real constant $c > 0$, $\exists n_0 \geq 1 : f(n) > cg(n) \geq 0, \forall n \geq n_0$.

Since $\mathcal{P}^{\tilde{\alpha}} - \mathcal{S} \subseteq \mathcal{D}$ and $\tau_n^* \geq \tilde{\tau}_n^{\tilde{\alpha}} \geq 0, \forall n \in \mathcal{N}$, we have that:

$$\sum_{n\in\mathcal{P}^{\tilde{\alpha}}-\mathcal{S}}\frac{\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}}{1+\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}} \leq \sum_{n\in\mathcal{D}}\frac{\beta p_n\tau_n^*}{1+\beta p_n\tau_n^*},$$

which gives us the lower bound on the average cache saving as $B^*(N) - \tilde{B}^{\alpha}(N) \geq |\overline{\mathcal{S}}^{\alpha}|$.

Recall the average cache occupancy defined in (3) and note that according to (16) for unpopular items we have $\tilde{\tau}_n^{\tilde{\alpha}} = 0, \forall n \notin \mathcal{P}^{\tilde{\alpha}}$. Also, according to Theorem 2, for static popular items we have $\tilde{\tau}_n^{\tilde{\alpha}} = \infty, \forall n \in \mathcal{S}^{\tilde{\alpha}}$. This gives:

$$\tilde{B}^{\tilde{\alpha}}(N) = \sum_{n=1}^{m(N)}\frac{\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}}{1+\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}} = |\mathcal{S}^{\tilde{\alpha}}| + \sum_{n\in\mathcal{P}^{\tilde{\alpha}}-\mathcal{S}}\frac{\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}}{1+\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}},$$

where $|\mathcal{P}^{\tilde{\alpha}} - \mathcal{S}| = m(N) - |\mathcal{S}^{\tilde{\alpha}}| \geq 0$. Using the solution given in (12), we have:

$$\begin{aligned}
\sum_{n\in\mathcal{P}^{\tilde{\alpha}}-\mathcal{S}}\frac{\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}}{1+\beta p_n\tilde{\tau}_n^{\tilde{\alpha}}} &= m(N) - |\mathcal{S}^{\tilde{\alpha}}| \\
- \sum_{n\in\mathcal{P}^{\tilde{\alpha}}-\mathcal{S}}&\frac{1}{\sqrt{1+2\frac{\beta c_f}{c_a\lambda_n}(p_n - p_{m(N)})}}.
\end{aligned} \tag{18}$$

Now, using the fact that:

$$\min_{\{\mathbf{x}\geq 0,\, \sum_{i=1}^{N} x_i = c\}}\sum_{i=1}^{N}\frac{1}{\sqrt{1+ax_i}} = \frac{N}{1+\frac{ac}{N}},$$

we can show that the second term in the right side of (18) is lower-bounded by:

$$\begin{aligned}
&\geq \frac{m(N) - |\mathcal{S}^{\tilde{\alpha}}|}{\sqrt{1+2\frac{\beta c_f}{c_a}\cdot\frac{1}{N^b - N^{a_1}}\sum_{n\in\mathcal{P}^{\tilde{\alpha}}-\mathcal{S}}\frac{p_n - p_{m(N)}}{\lambda_n}}} \\
&\geq m(N) - |\mathcal{S}^{\tilde{\alpha}}| - \frac{\beta c_f}{c_a\lambda_{min}},
\end{aligned}$$

where the second inequality comes from the fact that $\frac{1}{\sqrt{1+x}} \geq 1 - \frac{1}{2}x$ and $\sum_{n\in\mathcal{P}^{\tilde{\alpha}}-\mathcal{S}}\frac{p_n - p_{m(N)}}{\lambda_n} \leq \frac{1}{\lambda_{min}}$. Substituting the results gives the upper bound on the average cache occupancy completing part (ii) of the proof. ■

Theorem 3 reveals the potential of our proposed caching strategy which chooses items for caching based on their popularity and then incorporates the update rate of contents to decide how long each item should remain in the cache before eviction. Our proposed caching strategy completely discards the unpopular items, static or dynamic. More specifically, not caching the unpopular static items yields a very large gain on the cache saving side at a marginal loss on the average system cost side.

Theorem 3 shows that while the proposed strategy is asymptotically optimal for large data base sizes, it can also result in massive cache savings. This reveals that a cache size that grows with the rate of popular static items can achieve the same performance of having unconstrained cache size with the data base size being very large. As such, increasing the cache size beyond the threshold which is given as an upper bound in Theorem 3 will not reduce the average system cost for large data base sizes.

In the special scenario where the static items are unpopular for the given popularity measure $\tilde{\alpha}$, i.e., $\mathcal{S} \cap \mathcal{P}^{\tilde{\alpha}} = \phi$, Theorem 3 reveals that a *bounded cache size* of $\frac{\beta c_f}{c_a\lambda_{min}}$

can be asymptotically optimal and achieve the same average cost of a system with unconstrained cache size, even if the database size grows to infinity. Specifically, our proposed strategy is asymptotically optimal while massively reducing the cache occupancy to a constant cache size which does not grow with $N$.

Notice that the average cache occupancy for the unconstrained cache is not necessarily bounded by the order of popular static items. Not only does our proposed caching scheme achieve the same average cost of the system with unconstrained cache asymptotically but it also maintains a cache size which does not grow linearly with $N$. In other words, intelligently choosing the items to cache is a critical factor to optimize the average system cost in dynamic caching. If the popularity of static items is low, then caching only dynamic items considerably reduces the system's cost and attains remarkable cache space savings.

According to Theorem 3, $m(N)$ determines the trade off between how much cache storage is saved and how fast the cost converges to the optimal. Larger $m(N)$ will result in a faster convergence but a smaller cache saving gain. To make this trade-off more clear through an example, consider a set of items with Zipf(1) popularity distribution and assume $\lambda = \frac{1}{100}$ is equal for all the dynamic items. Consider $m(N) = N^b, 0 < b \leq 1$ to be the number of most popular items that will be considered for caching and also assume $|\overline{\mathcal{S}}^{\tilde{\alpha}}| = N^{0.5}$ is the number of static items which are not in the popular set. According to Theorem 3, the sufficient condition for asymptotic optimality is $b > \frac{1}{2}$. For such a choice of $m(N)$ we, investigate the trade-off. We adopt the percentage cost reduction of our proposed caching strategy for the constrained cache to the optimal solution derived for the unconstrained cache as our performance metric. Such a metric is defined as:

$$\text{Cost Reduction}(\%) = 100 \times \frac{\tilde{C}^{\tilde{\alpha}}(N) - C^*(N)}{C^*(N)}.$$

This percentage is depicted in Fig. 3 as a function of cache saving for different values of $N$. According to the figure, and as expected from Theorem 3, for any choice of $m(N) = N^b$ with $b > \frac{1}{2}$ as $N$ increases from 1000 to 10000, the proposed cost for the constrained cache converges to the optimal cost for an unconstrained cache size. The x-axis shows the amount of cache saving for the proposed strategy compared to the optimal average cache size for the unconstrained case. The figure illustrates that, as $N$ increases, the cache saving also increases, while the cost of the proposed policy converges to the optimal cost. This behavior demonstrates the potential of our proposed asymptotic strategy in massive cache savings. In addition, as $m(N)$ increases, the rate of convergence increases at the expense of having smaller savings in the cache size, as predicted by our theoretical result. In other words, smaller $m(N)$ result in bigger cache saving but with a slower convergence rate in cost. This is exactly the trade-off that theorem 3 revealed for the proposed asymptotic strategy.

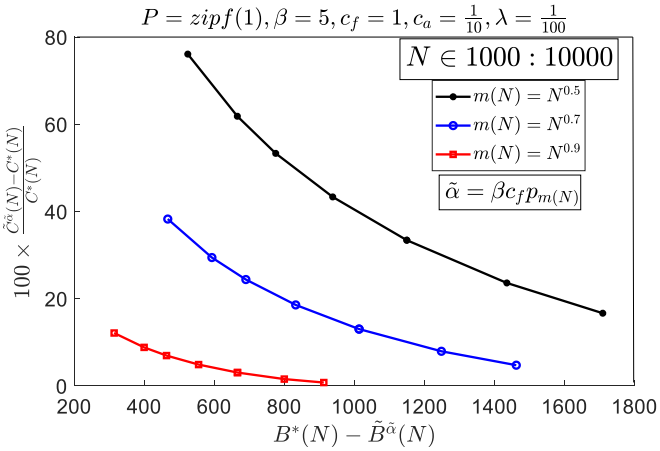


Fig. 3: Rate of convergence and cache saving trade-off

## VI. Conclusion

In this work, we have proposed and investigated an increasingly important caching scenario for serving dynamically changing content. We introduced the *age-of-version* metric to capture the served content's freshness and track the number of stale versions per content. We have addressed the problem of developing optimal caching strategies for minimizing the system's cost which is shaped by a combination of the service cost of fetching fresh content directly from a back-end database and the aging cost of cached, potentially older, content from a front-end cache. In the scenario of constrained cache size, our analysis have revealed the interesting fact that the optimal caching strategy allocates cache space to items based solely on their popularity, while the content update rate is what determines the content holding time in the cache. Moreover, we have explored the trade-off between the cost minimization and cache savings gain of our design. In particular, not only the cost of our proposed strategy converge asymptotically to the optimal strategy as the number of data items grows, but can also reduce the cache occupancy substantially, as fully characterized by our analysis and illustrated with numerical results.

## Appendix

### A. Proof of Lemma 3:

To check that Slater condition holds for any $0 < B < N$, assume $\tau_n = \frac{1}{2\beta p_n} \frac{B}{N-B} > 0, \forall n \in \mathcal{N}$ which gives $g(\boldsymbol{\tau}) < B$. So choose $\boldsymbol{\tau} = \frac{1}{2\beta} \frac{B}{N-B} (\frac{1}{p_1}, ..., \frac{1}{p_N}) \in \mathcal{F}_B$ which is a feasible point and all the inequalities are inactive.

To check the non-degeneracy assumption, we need to show that every where that a constraint is active, it's gradient is nonzero. Since constraints $\tau_n \geq 0, \forall n \in \mathcal{N}$ have always nonzero gradient, so we only need to check this for $g(\boldsymbol{\tau}) = \sum_{n=1}^{N} \frac{\beta p_n \tau_n}{\beta p_n \tau_n + 1} - B$. We have $\nabla g(\boldsymbol{\tau}) \neq \mathbf{0}$. To check the third condition, consider $\boldsymbol{\tau}' = (0, ..., 0) \in \mathcal{F}_B$ and choose $t_n = \frac{c}{n}$ such that $c\boldsymbol{\tau} \in \mathcal{F}_B$ for a given $\boldsymbol{\tau}$. Then for this choice of $\boldsymbol{\tau}'$ and $t_n$ we can show that condition 3 holds for all $\tau \in \mathcal{F}_B$. To check the last condition, notice that $L_C(\boldsymbol{\tau}) = \left\{\boldsymbol{\tau}' \in R^N : C(\boldsymbol{\tau}') < \mathrm{C}(\boldsymbol{\tau})\right\}$ is sub level set of the convex function $C(\boldsymbol{\tau})$ and therefore is also itself a convex set.

## REFERENCES


[1] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Wireless multicasting for content distribution: Stability and delay gain analysis," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1–9.

[2] ——, "Delay gain analysis of wireless multicasting for content distribution," *IEEE/ACM Transactions on Networking*, 2020.

[3] J. Zhang, "A literature survey of cooperative caching in content distribution networks," *arXiv preprint arXiv:1210.0071*, 2012.

[4] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–9.

[5] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[6] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1382–1393, 2016.

[7] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Achieving freshness in single/multi-user caching of dynamic content over the wireless edge," in *IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2020.

[8] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley, "Performance evaluation of hierarchical ttl-based cache networks," *Computer Networks*, vol. 65, pp. 212–231, 2014.

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

[10] D. S. Berger, P. Gland, S. Singla, and F. Ciucu, "Exact analysis of ttl cache networks," *Performance Evaluation*, vol. 79, pp. 2–23, 2014.

[11] J. Jung, A. W. Berger, and H. Balakrishnan, "Modeling ttl-based internet caches," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 1. IEEE, 2003, pp. 417–426.

[12] M. Dehghan, L. Massoulie, D. Towsley, D. S. Menasche, and Y. C. Tay, "A utility optimization approach to network cache design," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1013–1027, 2019.

[13] J. Zhong, R. D. Yates, and E. Soljanin, "Two freshness metrics for local cache refresh," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 1924–1928.

[14] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 2731–2735.

[15] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1897–1910, 2016.

[16] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger, "Age-optimal constrained cache updating," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 141–145.

[17] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 1681–1685.

[18] C. Kam, S. Kompella, and A. Ephremides, "Age of information under random updates," in *2013 IEEE International Symposium on Information Theory*. IEEE, 2013, pp. 66–70.

[19] J. Zhong, E. Soljanin, and R. D. Yates, "Status updates through multicast networks," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2017, pp. 463–469.

[20] E. Najm and R. Nasser, "Age of information: The gamma awakening," in *2016 IEEE International Symposium on Information Theory (ISIT)*. Ieee, 2016, pp. 2574–2578.

[21] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[22] 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)*. IEEE, 2017, pp. 136–140.

[23] D. Wessels, *Web caching*. " O'Reilly Media, Inc.", 2001.

[24] A. R. Cassandra, "Exact and approximate algorithms for partially observable markov decision processes," 1998.

[25] Q. Ho, "Necessary and sufficient kkt optimality conditions in non-convex optimization," *Optimization Letters*, vol. 11, no. 1, pp. 41–46, 2017.