Age-Optimal Constrained Cache Updating

†Roy D. Yates, ‡Philippe Ciblat, ♯Aylin Yener, ‡Michèle Wigger
† Rutgers University, USA, ryates@winlab.rutgers.edu
‡Telecom ParisTech, Université Paris-Saclay, France, ciblat@telecom-paristech.fr, wigger@telecom-paristech.fr
♯The Pennsylvania State University, USA, yener@engr.psu.edu

***Abstract*—We consider a system where a local cache maintains a collection of N dynamic content items that are randomly requested by local users. A capacity-constrained link to a remote network server limits the ability of the cache to hold the latest version of each item at all times, making it necessary to design an update policy. Using an age of information metric, we show under a relaxed problem formulation that an asymptotically optimal policy updates a cached item in proportion to the square root of the item's popularity. We then show experimentally that a physically realizable policy closely approximates the asymptotic optimal policy.**

I. Introduction

Consider a local cache connected by a capacity-constrained link to a remote network server, as shown in Figure 1. The local cache maintains a collection of N dynamic content items that are randomly requested by local users. These content items are dynamic in that they are subject to random continuous replacement by newer versions at the remote server. However, the capacity-constrained remote link limits the ability of the local cache to maintain the latest version of each item. When a user requests a content item, the cache sends its local (possibly outdated) version.

We observe that this system model arises in various settings. For example, the content items could be records in a database and the cache provides access to a local copy. The cache could also represent a small-cell base station that delivers popular content to nearby mobile users. Downloads by the mobile users would occur over short-range links in unlicensed spectrum but the remote link would be over a longer distance and in licensed spectrum. In this case, a network operator will wish to limit the rate of the remote link. In another scenario, the local caches may be local storage in a TV/video news distribution system with the remote link by a satellite that broadcasts the same content updates to thousands of local caches.

In these examples, the remote link is slow and/or expensive relative to the links from the cache to the local users; this precludes going to the remote server to satisfy each local request. Item downloads to the local users will occur at a much higher frequency than downloads from the remote link to the cache. Thus our goal is to try to ensure that the items delivered by the cache to the local users are as up-to-date as possible. Because the remote server has the current version of each item, the question is how should updated items be downloaded from the server. Put another way, how should the remote server send updates to the local cache so that users receive the most recent versions of the items they request?

To examine this question, Section II develops a simple discrete-time model in which only one update can occur in each slot. Based on an *age of information* metric to measure how outdated are the cache's responses to locally requested items, we formulate a finite horizon update scheduling problem to minimize the average age of a requested item. A relaxation of this problem then provides a lower bound on the average age. In Section III, we identify a simple asymptotic solution to the relaxed problem. Since these steps relax the discrete-time updating model, Section IV shows that a discrete-time approximation of the optimal asymptotic policy suffers only a small penalty. Performance results comparing analysis and simulation appear in Section V. Section VI concludes the paper.

A. Related Work

In applying an age of information (AoI) metric to updates in a cache, this paper is related to several recent works on AoI; see, for example, [1]–[3]. These works employ the canonical setting for AoI analysis: one or more sources send status updates about processes to a monitor across a network service facility. In this work, the remote server is the source, the dynamic content items are the processes, and the local cache is the monitor. As in [4], [5], multiple processes are being updated simultaneously. However, in [5] each process is associated with an independent updating source that is competing for the service facility. Instead, the remote server here is a single source that controls the updating of all processes. This work also has some similarity to [6]–[8] in that updates are not submitted as a Poisson process; rather, the source can control the submission of updates. However, these prior contributions control the updates to mitigate the impact of randomness in the service times. In this work, the service time model is simply that transmission of an update requires a deterministic unit of service. With respect to AoI, the primary novelty of this work is the use of a popularity-weighted AoI metric. The goal is to minimize the average age of items requested from the local cache. Thus the AoI of an item is weighted by how frequently it is requested.

II. Problem Formulation

We assume the cache hosts a set of N timestamped items $\mathcal{N} = \{1, \ldots, N\}$. Because the items are subject to version updates, we use $X_n(t)$ to track the age of item n at time t. That is, if the version of item n in the cache at time t is timestamped $v_n(t)$ then $X_n(t) = t - v_n(t)$.

978-1-5090-4096-4/17/$31.00

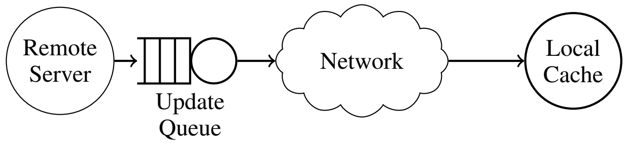

Fig. 1. Local cache system.

We assume time is slotted such that the transmission of an update from the server to the cache requires exactly one slot. For an integer t, slot t refers to the unit time interval $(t-1, t]$. When item n is updated in slot t_0, a version of item n with timestamp $t_0 - 1$ begins transmission at time $t_0 - 1$ and becomes available for downloading from the cache at time t_0. At time t_0 the age $X_n(t)$ is reset to $X_n(t_0) = 1$. If no subsequent updates of item n occur in the time interval (t_0, t_1) then $X_n(t_1) = t_1 - (t_0 - 1)$. For convenience of exposition, we assume every item is updated in slot zero so that $X_n(0) = 1$ for every item n. This yields the discrete-time sawtooth process depicted in Figure 2.

Because the link to the remote server is constrained, the local cache is able to download items at a rate λ, a fraction of the rate at which items are delivered to local users. In time slot t, a single file u_t may be updated from the server. We use $u_t = 0$ to mark a slot t in which no file is updated. The update vector $\mathbf{u} = [u_1, \ldots, u_T]$ then specifies the age process $\{X_n(t) : 0 \leq t \leq T\}$ for each item n.

The items the cache hosts have varying degrees of popularity. In particular, item n is requested from the cache with probability $p_n > 0$, independent of all other requests. We refer to p_n as the *popularity* of item n and to

$$\mathbf{p} = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix} \tag{1}$$

as the *popularity vector*.

Given the update vector $\mathbf{u}$, a request for item n made at a time uniformly distributed over $[0, T]$ has average age

$$\overline{X}_n(\mathbf{u}) = \frac{1}{T} \int_0^T X_n(t)\, dt. \tag{2}$$

Based on the popularity model, the average age of a randomly requested item from the cache is

$$\overline{X}(\mathbf{u}) = \sum_{n=1}^{N} p_n \overline{X}_n(\mathbf{u}). \tag{3}$$

Our objective is to find the update sequence $u_1, \ldots, u_T$ that minimizes $\overline{X}(\mathbf{u})$, i.e., we would like to solve the following update scheduling problem:

$$\overline{X}^*(K, T) = \min_{\mathbf{u}} \overline{X}(\mathbf{u}). \tag{4a}$$

subject to

$$u_t \in \{0, \ldots, N\} \quad \text{for all } t, \tag{4b}$$

$$\sum_{t=1}^{T} 1_{\{u_t > 0\}} = K. \tag{4c}$$

As $1_{\{A\}} = 1$ if event A is true, and is otherwise zero, the constraint (4c) limits the remote server to K updates over the T slots. Furthermore, implicit in the definition of the update vector $\mathbf{u}$ are the following constraints:

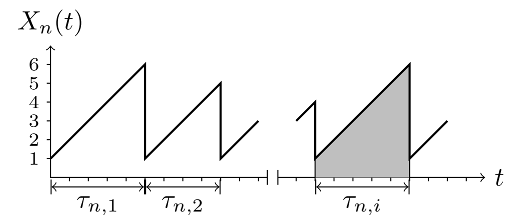

Fig. 2. The discrete time age process $X_n(t)$, with updates at times $t_1 = 5$ and $t_2 = 9$. The inter-update times are $\tau_1 = 5$, and $\tau_2 = 4$.

C1: Updates occur only in discrete time slots.

C2: Updates are collision-free; no more than one item is updated in a slot.

While these constraints suggest (4) is an intractable combinatorial optimization problem, the complexity class of (4) remains open. We sidestep this question with the following approach:

1) In Section II-A, we relax constraints C1 and C2 and express the relaxed optimization problem in terms of the inter-update times $\{\tau_{n,i}\}$. In this relaxation, the transmission of an update is not limited to an integer slot time and that multiple items can be updated simultaneously.
2) In Section III-A, we assume that k_n updates are allocated to item n, and find the age-minimizing update schedule for item n.
3) Finally, in Section III-B, we optimize the allocation $k_1, \ldots, k_N$ of updates across all N items in the limit of large T; integer optimization of k_n is replaced by continuous optimization of the asymptotic update rate $\lambda_n = \lim_{T\to\infty} k_n/T$.

The relaxation of constraints C1 and C2 leads to an approximate problem that provides a lower bound on $\overline{X}^*(K, T)$, denoted by $\overline{X}^*_{lb}(K, T)$. In the asymptotic setting of large T, this lower bound is optimized to yield $\overline{X}^*_{alb}(\mathbf{p}, \lambda)$, a function of the popularity vector $\mathbf{p}$ and the overall update rate $\lambda = \sum_{n=1}^{N} \lambda_n$.

A. Relaxed Problem on Inter-Update Times

We start by reformulating (and then relaxing) Problem (4). Given updates of item n occur in slots $t_{n,0} = 0, t_{n,1}, \ldots, t_{n,k_n}$, we define the inter-update times as

$$\tau_{n,i} = \begin{cases} t_{n,i} - t_{n,i-1}, & i = 1, \ldots, k_n, \\ T - t_{n,k_n}, & i = k_n + 1. \end{cases} \tag{5}$$

Referring to Figure 2, the average of $X_n(t)$ in (2) can be expressed as a function of the vector $\boldsymbol{\tau}_n = (\tau_{n,1}, \ldots, \tau_{n,k_n+1})$ of inter-update times as

$$\overline{X}_n(\boldsymbol{\tau}_n) = \frac{1}{T} \sum_{i=1}^{k_n+1} \int_0^{\tau_{n,i}} (t+1)\, dt = \sum_{i=1}^{k_n+1} \frac{\tau_{n,i}^2 + 2\tau_{n,i}}{2T}. \tag{6}$$

Since $\sum_{i=1}^{k_n+1} \tau_{n,i} = T$, it follows from (6) that

$$\overline{X}_n(\boldsymbol{\tau}_n) = \frac{1}{2T} \sum_{i=1}^{k_n+1} \tau_{n,i}^2 + 1. \tag{7}$$

We can reformulate Problem (4) by expressing the constraint C2 in terms of $\tau_{n,i}$; however, this fails to simplify the problem. Instead, the relaxation of both constraints C1 and C2 leads to the following approximate problem providing a lower bound ~~of~~on $\overline{X}^*(K,T)$, denoted by $\overline{X}^*_{lb}(K,T)$.

$$\overline{X}^*_{lb}(K,T) = \min_{\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_N} \sum_{n=1}^{N} p_n \overline{X}_n(\boldsymbol{\tau}_n) \tag{8a}$$

subject to

$$\tau_{n,i} \geq 0, \text{ for all } n, i, \tag{8b}$$

$$\tau_{n,1} + \cdots + \tau_{n,k_n+1} = T, \text{ for all } n, \tag{8c}$$

$$\sum_{n=1}^{N} k_n = K \text{ with } k_n \in \mathbb{N}. \tag{8d}$$

III. Relaxed Problem Resolution

To solve Problem (8), we have to find the number of updates k_n and the inter-update times $\boldsymbol{\tau}_n$ for all n. Given k_n, however, Problem (8) becomes separable. Therefore we can work item by item, minimizing $\overline{X}_n(\boldsymbol{\tau}_n)$ in (7) for each item n.

A. Optimal Inter-Update Times for Each Item

For fixed k_n, our relaxed subproblem for item n is:

$$\overline{X}_{lb,n}(\boldsymbol{\tau}^*_n) = \min_{\boldsymbol{\tau}_n} \overline{X}_n(\boldsymbol{\tau}_n) \tag{9a}$$

subject to

$$\tau_{n,i} \geq 0 \text{ for all } i, \tag{9b}$$

$$\tau_{n,1} + \cdots + \tau_{n,k_n+1} = T. \tag{9c}$$

According to (7), this optimization problem is convex. From the Karush-Kuhn-Tucker (KKT) conditions, we obtain

$$\tau^*_{n,i} = \frac{T}{k_n+1}, \qquad i = 1, \ldots, k_n+1. \tag{10}$$

This verifies that identical deterministic inter-update times is an optimal policy for a limited number of updates of a single item. Applying (10) to (7), this policy yields

$$\overline{X}_{lb,n}(\boldsymbol{\tau}^*_n) = \overline{X}^*_{lb,n}(k_n) = \frac{T}{2(k_n+1)} + 1. \tag{11}$$

B. Optimal Update Frequencies

According to (8) and (11), we now would like to solve the following optimization problem over the k_n.

$$\overline{X}^*_{lb}(K,T) = \min_{k_1,\ldots,k_N} \sum_{n=1}^{N} p_n \overline{X}^*_{lb,n}(k_n) \tag{12a}$$

subject to

$$k_n \in \mathbb{N}, \text{ for all } n, \tag{12b}$$

$$\sum_{n=1}^{N} k_n = K. \tag{12c}$$

Although (12) is substantially simpler than (8), it remains hard to solve due to the integer assumption on k_n. To overcome this issue, we will consider an asymptotic approach (i.e., large T.) We replace k_n with $\lambda_n = k_n/T$, where λ_n is now a positive real-valued term. In the limit of large T, the relaxed average age of item n in (11) becomes[1]

$$\overline{X}^*_{alb,n}(\lambda_n) = \lim_{T\to\infty} \overline{X}^*_{lb,n}(k_n) = \frac{1}{2\lambda_n} + 1. \tag{13}$$

Problem (12) is thus reformulated as

$$\overline{X}^*_{alb}(\mathbf{p},\lambda) = \min_{\lambda_1,\ldots,\lambda_N} \sum_{n=1}^{N} p_n \overline{X}^*_{alb,n}(\lambda_n) \tag{14a}$$

subject to

$$\lambda_n \geq 0, \tag{14b}$$

$$\lambda_1 + \cdots + \lambda_N = \lambda. \tag{14c}$$

We note that the subscript "a" stands for *asymptotic*. Given (13), one can readily see that optimization problem (14) is convex. From its KKT conditions, we obtain

$$\lambda^*_n = \frac{\lambda\sqrt{p_n}}{\sum_{i=1}^{N}\sqrt{p_i}}. \tag{15}$$

We remark the update rate of item n follows a square-root law with respect to its popularity. Applying (13) and (15) to (14), we find that the minimum asymptotic relaxed average age is

$$\overline{X}^*_{alb}(\mathbf{p},\lambda) = \frac{\Delta^*(\mathbf{p})}{\lambda} + 1, \tag{16}$$

with

$$\Delta^*(\mathbf{p}) = \frac{1}{2}\Big(\sum_{i=1}^{N}\sqrt{p_i}\Big)^2. \tag{17}$$

Since $\overline{X}^*_{alb}(\mathbf{p},\lambda)$ was obtained by relaxing the integer-assumption on the inter-update times, it forms a lower bound on the average age of a requested item under the optimal collision-free discrete-time schedule. Defining $\overline{X}^*_a(\mathbf{p},\lambda) = \lim_{T\to\infty}\overline{X}^*(\lambda T, T)$, we have

$$\overline{X}^*_{alb}(\mathbf{p},\lambda) \leq \overline{X}^*_a(\mathbf{p},\lambda). \tag{18}$$

C. Discussion

We observe in (15) that the update rate for item n should be proportional to the square root of its popularity. "Square root" proportional caching policies have appeared before in [9]. In this work, each item was a file that was mirrored at a random number of sites in a content distribution network (CDN). Each site could hold only a fraction of the entire collection of files, so a user requesting a file would randomly search sites in the CDN until the file was found. It was shown that to minimize the average number of sites searched, the fraction of sites that host file n should be in proportion to the square root of the popularity of file n. In the CDN, the item/file was sprinkled over the sites and there was a linear cost in how many sites must be visited to find the item. In version caching, item updates are sprinkled through time and there is a linear cost in how far back in time one must go to find the most recent item update. In both cases, items (or item updates) are sprinkled in proportion to the square root of their popularity.

[1]As $\lambda_n = \lim_{T\to\infty} k_n/T$, we also have $\lambda_n = \lim_{T\to\infty}(k_n+1)/T$.

IV. Discrete-Time Updating

We now describe a practical update policy that takes into account constraint C1 and approximates the ideal periodic updating that led to $\overline{X}^*_{alb}(\mathbf{p}, \lambda)$ derived in the previous section. Our approach is to update item n with randomly quantized i.i.d. inter-update times $Z_{n,1}, Z_{n,2}, \ldots$. Defining $\overline{\tau}_n = \lceil \tau^*_n \rceil$ and $q_n = \overline{\tau}_n - \tau^*_n$ (and so $q_n \in [0,1]$), each $Z_{n,i}$ is defined to have PMF

$$P_{Z_n}(z) = \begin{cases} q_n & z = \overline{\tau}_n - 1, \\ 1 - q_n & z = \overline{\tau}_n, \\ 0 & \text{otherwise}. \end{cases} \tag{19}$$

It is straightforward to verify that

$$\mathbb{E}[Z_n] = \tau^*_n \qquad \mathbb{E}\left[Z_n^2\right] = [\tau^*_n]^2 + q_n(1-q_n). \tag{20}$$

Under the update policy Z_n, the process $X_n(t)$ is as shown in Figure 2, with inter-update times $Z_{n,i} = \tau_{n,i}$. The value of $\overline{X}_n(\mathbf{u})$ in (2) can be viewed as the average reward rate of a renewal-reward process [10]. Each update marks a renewal and renewal i earns a reward $R_i = Z^2_{n,i}/2 + Z_{n,i}$ equal to the integral of $X_n(t)$ over the renewal period. (In Figure 2, the shaded area depicts the reward R_i.) By the renewal-reward theorem, the average reward rate of this discrete-time (dt) updating policy is

$$\overline{X}_{adt,n} = \lim_{T\to\infty} \overline{X}_n(T) = \frac{\mathbb{E}\left[Z_n^2/2 + Z_n\right]}{\mathbb{E}[Z_n]} = \frac{\mathbb{E}\left[Z_n^2\right]}{2\mathbb{E}[Z_n]} + 1. \tag{21}$$

Applying (20), we obtain

$$\begin{aligned} \overline{X}_{adt,n} &= \frac{\tau^*_n}{2} + \frac{q_n(1-q_n)}{2\tau_n} + 1 \\ &= \frac{1}{2\lambda^*_n} + \frac{q_n(1-q_n)\lambda^*_n}{2} + 1. \end{aligned} \tag{22}$$

It follows from (15) that the average age of a requested item is

$$\begin{aligned} \overline{X}_{adt}(\mathbf{p}, \lambda) &= \sum_{n=1}^{N} p_n \overline{X}_{adt,n} \\ &= \overline{X}^*_{alb,n}(\mathbf{p}, \lambda^*) + \frac{1}{2}\sum_{n=1}^{N} p_n q_n (1-q_n)\lambda^*_n. \end{aligned} \tag{23}$$

We define the quantization error associated with quantizing the inter-arrival times to integer numbers of slots as

$$\varepsilon = \frac{1}{2}\sum_{n=1}^{N} p_n q_n(1-q_n)\lambda^*_n. \tag{24}$$

Since $q_n(1-q_n) \leq 1/4$, it follows from (24) that $\varepsilon \leq \lambda\overline{\varepsilon}(\mathbf{p})$ where

$$\overline{\varepsilon}(\mathbf{p}) = \frac{1}{8}\sum_{n=1}^{N} p_n \lambda^*_n = \frac{1}{8}\frac{\sum_{n=1}^{N} p_n^{3/2}}{\sum_{i=1}^{N} p_i^{1/2}}. \tag{25}$$

Thus, by (18), we obtain the inequality

$$\overline{X}^*_{alb}(\mathbf{p}, \lambda) \leq \overline{X}_{adt}(\mathbf{p}, \lambda) \leq \overline{X}^*_{alb}(\mathbf{p}, \lambda) + \lambda\overline{\varepsilon}(\mathbf{p}). \tag{26}$$

As $p_n \leq 1$, we get $p_n^{3/2} \leq p_n^{1/2}$ which implies that $\overline{\varepsilon}(\mathbf{p}) \leq 1/8$. As each item transmission requires one time slot, an extra $\lambda/8$ of a time slot on the average is essentially negligible.

V. Examples and Simulations

A. Popularity Distributions

For our experiments, we compare uniform (equal) and Zipf popularity distributions.

Uniform Popularity: In this case, $p_n = 1/N$, implying $\lambda^*_n = \lambda/N$, $\tau^*_n = N/\lambda$ and

$$\Delta^*(\mathbf{p}) = \frac{N}{2} \qquad \overline{\varepsilon}(\mathbf{p}) = \frac{1}{8N}. \tag{27}$$

Uniform popularity induces round robin updating of each item every N slots. Note that the age performance of uniform popularity is achieved by any popularity vector $\mathbf{p}$ when the items are updated in round robin fashion.

Zipf Popularity: For the Zipf distribution, $p_n = C_N(s)/n^s$ where $s \geq 0$and

$$C_N(s) = \Big(\sum_{n=1}^{N} \frac{1}{n^s}\Big)^{-1}. \tag{28}$$

From (17) and (25),

$$\Delta^*(\mathbf{p}) = \frac{C_N(s)}{2C^2_N(s/2)}, \qquad \overline{\varepsilon}(\mathbf{p}) = \frac{C_N(s)C_N(s/2)}{8C_N(3s/2)}. \tag{29}$$

Thanks to Cauchy-Schwartz inequality, one can prove that $\Delta^*(\mathbf{p})$ for Zipf popularity is less than $N/2$ and so better than the uniform popularity case.

B. Collision-Resolution Mechanism

In the development of the asymptotic lower bound, we have ignored the effect of colliding updates on the network link. This issue can be resolved by queuing. Unlike prior work, [1]–[5] for example, in which queueing contributed substantially to the age, a smart queuing technique for the problem at hand makes that the performance loss related to queueing is practically nil. We propose the following mechanism. If an update is scheduled in slot t for which there is already a backlog of b queued updates, then that update will be deferred to slot $t+b$. However, the server does *not* queue the slot t version of the item. Instead, in slot $t+b$, the server sends the current slot $t+b$ version. The scheduling queue thus does not cause delivered updates to become stale, but rather adds some additional randomness to the inter-update times for each item.

C. Simulation Experiments

In our simulation experiments, we start with a given popularity vector $\mathbf{p}$ and overall update rate λ. The update rates $\lambda^*_n = 1/\tau^*_n$ are calculated using (15). For each item n, a randomly quantized schedule of i.i.d. inter-update times $Z_{n,i}$ based on (19) is then generated and the collision resolution mechanism described in the previous subsection is applied. The resulting system generates a slot-quantized collision-free update schedule that was required by the original update scheduling problem (4).

Except otherwise stated, we fix $N = 50$ items, and a simulation time of $T = 100 \max_n \tau^*_n$ slots, which means the least popular item is updated roughly 100 times. In each figure, solid lines represent the closed-form asymptotic lower

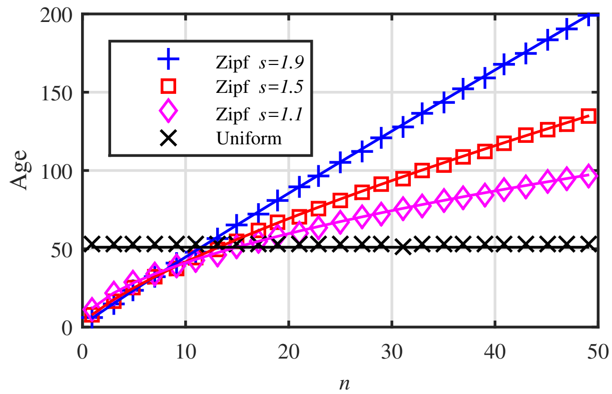

Fig. 3. The average age $\overline{X}^*_{alb,n}(\mathbf{p}, \lambda = 0.5)$ for each item n.

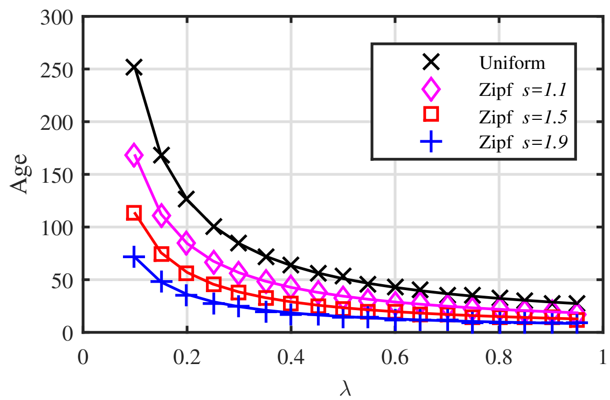

Fig. 4. The average age $\overline{X}^*_{alb}(\mathbf{p}, \lambda)$ as a function of λ.

bound expressions obtained in Section III whereas the markers correspond to the evaluations through the experimental protocol described above.

For Zipf-distributed and uniform popularity vectors, Figure 3 plots the average age for each item n with overall update rate $\lambda = 0.5$. In the figure, solid lines depict $\overline{X}^*_{alb,n}(\mathbf{p}, \lambda)$ in (16). As expected, the proposed policy reduces the average age of more popular items at the expense of less popular items. Moreover, as the solid lines are close to the markers, the effects of quantization and the scheduling queue are negligible for all items, independent of an item's popularity.

In Figure 4, we examine the average age (averaged over all items) as a function of the overall update rate λ. The solid lines show the asymptotic lower bound average age $\overline{X}^*_{alb}(\mathbf{p}, \lambda)$ for each popularity vector $\mathbf{p}$. We observe that at all arrival rates λ, the experimental protocol essentially matches the corresponding asymptotic lower bound. This figure also shows that as the Zipf parameter s increases, optimization of the update rates is able to exploit the concentration in the popularities.

In Figure 5, we show how the normalized average age $\Delta^*(\mathbf{p})$ varies with popularity for $N = 50$ items. Specifically, for Zipf popularity, we see that increasing parameter s reduces the average age as the popularity distribution concentrates on fewer but more popular items. Similarly, as $s \to 0$ and Zipf approaches uniform, the average age approaches that of uniform popularity.

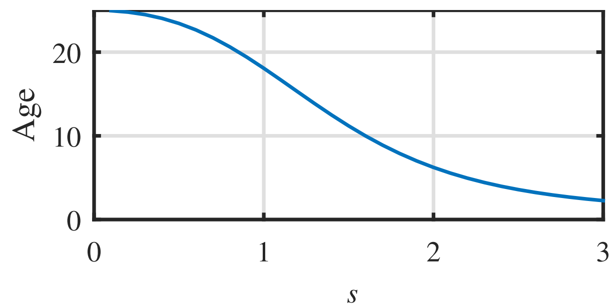

Fig. 5. The normalized average age $\Delta^*(\mathbf{p})$ for $N = 50$ items with Zipf (s) popularity as a function of parameter s.

VI. Conclusion

This work introduces a popularity-weighted age of information metric for updating dynamic content in a local cache. We have shown that updating rates of the items in the local cache should be proportional to the square-root of items' popularity among the requesting users. Finding a tractable solution to the optimal update schedule has required an asymptotic treatment. We have observed that the performance of a physical realizable policy is indistinguishable from that of the asymptotic one.

An interesting future research direction for the proposed framework is to consider item popularities that evolve in time and are dependent on previous requests. This allows modeling scenarios where a file can become more popular the more often it is downloaded and viewed.

VII. Acknowledgement

This work was supported in part by NSF award CIF-1422988.

References

[1] S. Kaul, R. D. Yates, and M. Gruteser. Real-time status: How often should one update? In *Proc. INFOCOM*, 2012.

[2] M. Costa, M. Codreanu, and A. Ephremides. On the age of information in status update systems with packet management. *IEEE Trans. Info Theory*, 62(4):1897–1910, April 2016.

[3] C. Kam, S. Kompella, G. D. Nguyen, and A. Ephremides. Effect of message transmission path diversity on status age. *IEEE Trans. Info Theory*, 62(3):1360–1374, March 2016.

[4] R. D. Yates and S. Kaul. Real-time status updating: Multiple sources. In *Proc. IEEE Int'l. Symp. Info. Theory*, July 2012.

[5] R. D. Yates and S. K. Kaul. The age of information: Real-time status updating by multiple sources. *CoRR*, abs/1608.08622, 2016.

[6] R. D. Yates. Lazy is timely: Status updates by an energy harvesting source. In *Proc. IEEE Int'l. Symp. Info. Theory*, 2015.

[7] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff. Update or wait: How to keep your data fresh. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[8] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff. Update or wait: How to keep your data fresh. *CoRR*, abs/1601.02284, 2016. Submitted to IEEE Trans. Info. Theory.

[9] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. In *ACM SIGCOMM Computer Communication Review*, volume 32, pages 177–190. ACM, 2002.

[10] R. G. Gallager. *Stochastic processes: theory for applications*. Cambridge University Press, 2013.