Age-Based Cache Updating Under Timestomping

Subhankar Banerjee Priyanka Kaswan Sennur Ulukus
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
sbanerje@umd.edu *pkaswan@umd.edu* *ulukus@umd.edu*

***Abstract*—We consider a slotted communication system consisting of a source, a cache, a user and a timestomping adversary. The time horizon consists of total T time slots, such that the source transmits update packets to the user directly over T_1 time slots and to the cache over T_2 time slots. We consider $T_1 \ll T_2$, $T_1 + T_2 < T$, such that the source transmits to the user once between two consecutive cache updates. Update packets are marked with timestamps corresponding to their generation times at the source. All nodes have a buffer size of one and store the packet with the latest timestamp to minimize their age of information. In this setting, we consider the presence of an oblivious adversary that fully controls the communication link between the cache and the user. The adversary manipulates the timestamps of outgoing packets from the cache to the user, with the goal of bringing staleness at the user node. At each time slot, the adversary can choose to either forward the cached packet to the user, after changing its timestamp to current time t, thereby rebranding an old packet as a fresh packet and misleading the user into accepting it, or stay idle. The user compares the timestamps of every received packet with the latest packet in its possession to keep the fresher one and discard the staler packet. If the user receives update packets from both cache and source in a time slot, then the packet from source prevails. The goal of the source is to design an algorithm to minimize the average age at the user, and the goal of the adversary is to increase the average age at the user. We formulate this problem in an online learning setting and provide a fundamental lower bound on the competitive ratio for this problem. We further propose a deterministic algorithm with a provable guarantee on its competitive ratio.**

I. Introduction

To improve data availability and scalability in wireless networks, caches are often deployed at various network nodes, such as base stations and access points [1]. Caches are temporary storage units that hold copies of data that have been requested or are anticipated to be requested by the users. By serving frequently accessed data from caches, the original sources are relieved of handling every single request, allowing them to handle a larger number of concurrent users.

In settings where the requested content is dynamic in nature, we need to ensure that the cached data remains fresh by frequently updating the cache. In this work, we use the age of information metric to characterize the staleness of data. If data packet present at a node at time t was generated from the source at time $u(t)$, then the instantaneous age of information at the node is $t-u(t)$, see [2]–[4]. The goal of cache updating systems is to minimize the age of information at the end users.

In this respect, we consider the discrete-time cache updating system shown in Fig. 1, where we examine the evolution of the age of information at the user node over the time horizon T. Due to resource constraints, it is not feasible for the source to update the user in every time slot. The source is able to update the user node directly over T_1 time slots and the cache node over T_2 time slots, such that $T_1 \ll T_2$ and $T_1 + T_2 < T$. We assume that if the source sends an update to the cache in a time slot, it is possible for the cache to relay the packet to the user in the same time slot, i.e., packet can reach from source to user both directly or through cache within one time slot. For some of the results in this paper to go through, we assume that the source can transmit only once to the user directly between two consecutive source to cache updates. A more general version of the problem can be an interesting future work.

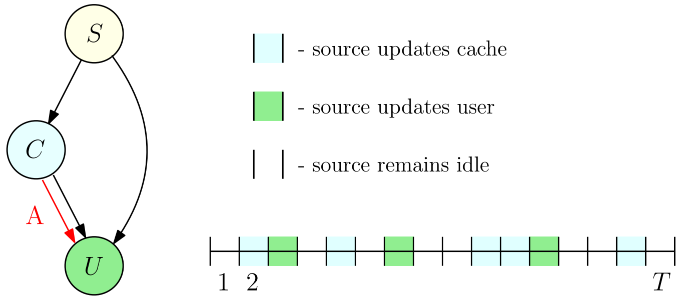

Fig. 1. System model for caching system.

In the absence of any adversary in this system, two observations can be made about the updating policy that should be adopted in this system. First, whenever the cache gets updated, then instead of waiting, the cache should immediately relay the update to the user in the same time slot, which will cause the age of the user to drop to 1 in the very next time slot. Second, both direct update and cache-aided updates have the same impact on age of information at the user, with age dropping to 1 in the following time slot. Hence, from [5], the user should receive an update, from either the cache or the source, at constant time intervals, with inter-update intervals of size $\frac{T}{T_1+T_2+1}$ to minimize its average age.

We next consider the presence of a timestomping adversary at the outgoing link of the cache which aims to deteriorate the average age at the user, through timestamp manipulation. If the user possesses a more recent packet than the packet present at the cache, the adversary would be inclined to send a copy of the cached packet to the user after changing its timestamp to current time t. The manipulated timestamp would trick the user into thinking that the received packet is fresher, making it discard its own packet in favor of the cached packet. On the other hand, if the user's packet is staler than the cached packet, than the adversary would be inclined to not send the cached packet, in other words, stay idle.

In the recent literature, multiple studies have been conducted on the age of information in the presence of an adversary, see [6]–[16]. All these works consider either an adversary that completely eliminates the update packet [9]–[14] or an adversary that decreases the signal to noise ratio of a communication link [6], [7] or an adversary that blocks the communication channel for a duration of time which results in higher age for the communication network [8] or an adversarial gossip network [15], [16]. Different than all this work, we consider an adversary that changes the time stamp of an update packet.

In [17], the authors have considered a timestomping adversary. Traditionally, timestomping is used by malware adversaries to make malicious files appear to be out of an attack timeframe and consequently bypass detection. [17] introduces how age-based cache updating systems are uniquely vulnerable to timestomping based attacks. This is because, timestomping alters the timestamps of packets, and in age-based systems, the decision to accept or reject an incoming packet is based on the comparison of timestamps of various packets.

In [17], the timestomping adversary attacks one node out of a gossip network of n nodes, and probabilistically alters the timestamps of all the incoming and outgoing packets of the attacked node, with the goal of introducing staleness at all nodes of the network. In [17], all nodes push updates according to a Poisson process with fixed rates in the continuous time setting and the adversary increases or decreases the timestamps probabilistically, without knowing the age or timestamps at other nodes. Different from [17], in this work, we consider a discrete time system where the adversary can transmit the cached update packet to the user at any time slot over the time horizon T and the adversary has the full knowledge of the packet transmission policy employed by the source and the timestamp at the user, which allows the adversary to choose its actions to transmit or not in every time slot wisely.

In this paper, we formulate the timestomping adversarial cache update problem as an online learning problem and study the competitive ratio for this problem. We assume the adversary is oblivious in nature, i.e., the adversary only has knowledge about the source transmission policy, however, it is required to generate the adversarial action for each timeslot over the entire time horizon T before the beginning of the first time slot. In this paper, we first provide a deterministic policy and show that this policy is $\left(\frac{1+T+T_1}{1+T_2+T_1+T}\right)\left(1+\frac{T_2}{T_1+1}\right)$ competitive. Then, we use Yao's minimax theorem [18] to find a fundamental lower bound on the competitive ratio for the considered system model.

II. System Model and Problem Formulation

We consider a wireless communication network where a source aims to minimize the age of a user with timely delivery of update packets to it. Due to power constraints, the source can only transmit update packets directly to the user for T_1 time slots over the time horizon of T. There is a cache in the system, and the source can transmit update packets to the cache for T_2 time slots over the same time horizon of T. As the transmission cost from the source to the cache is typically less than the transmission cost from the source to the user, we have $T_1 \ll T_2$. In addition, due to the power constraint of the source $T_1 + T_2 < T$. Without loss of generality, let $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{N}$. At a given time slot, the source can only transmit to either the user or to the cache. Each update packet contains its generation time as a time stamp. The cache and the user only keep their respective freshest update packets and discard any staler update packets.

There is also an adversary in the system. The adversary completely controls the communication link between the cache and the user, however, it cannot control the link between the source and the user. Whenever the user receives an update packet from the cache, it compares its time stamp with the time stamp of the already existing packet. If the packet from the cache has relatively fresher time stamp, the user accepts it and discards the existing packet. The adversary can transmit the cached update to the user at any time slot, i.e., there is no power constraint for the adversary. At any given time slot t, the age of the user is defined as $v(t) = t - u(t)$, where $u(t)$ is the generation time of the freshest packet that the user had before time slot t. At time t, if the user receives an update packet directly from the source, and also a cached update packet, then the user always accepts the packet from the source and discards the packet from the cache.

The adversary can manipulate the age of the user by changing the time stamp of the cached packet. For example, at time t, let the time stamp of the packet at the user be $t-2$, and the time stamp of the packet at the cache be $t-4$. The adversary may change the time stamp of the cached packet to t while transmitting it to the user. Upon receiving this packet, the user discards the fresher existing packet and accepts the staler packet which results in an increment of age of the system. Thus, the adversary can choose one of two actions, namely, it can either change the time stamp of the cached packet as the freshest packet, i.e., it changes the time stamp of the cached packet to current time t, or it can change the time stamp of the cached packet as the stalest packet, i.e., it changes the time stamp of the cached packet to the oldest time 0. We denote the former action as 1 and the latter action as 0. For time horizon T, we define a sequence σ as the adversarial sequence where $\sigma \in \{0,1\}^T$. A pictorial representation of the system model is given in Fig. 1.

The goal of the source is to design a transmission policy π to reduce the average age of the system, while the goal of the adversary is to design an adversarial sequence σ to increase the age of the system. We consider an oblivious adversary, i.e., the adversary has to generate the whole sequence σ at time 0. We define the average age of the system corresponding to a transmission algorithm π and an adversarial sequence σ as,

$$A^{\pi,\sigma} = \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_\pi[v^{\pi,\sigma}(t)] \quad (1)$$

In this work, to analyze the performance of an online algorithm

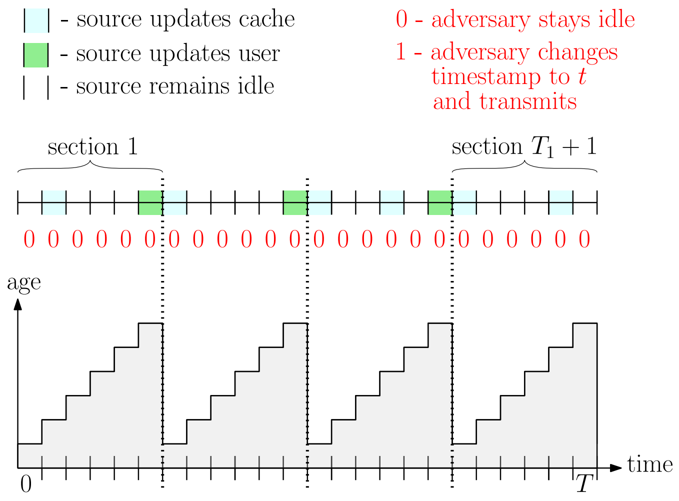

Fig. 2. Suggested deterministic algorithm, dividing timelines into $T_1 + 1$ sections of equal size. $T_1 = 3$, $T_2 = 6$, $T = 24$.

we use the competitive ratio metric [19]. The competitive ratio for any online algorithm π is defined as,

$$c^{\pi} = \sup_{\sigma} \frac{A^{\pi,\sigma}}{A^{o,\sigma}} \tag{2}$$

where superscript o stands for the optimal algorithm for the adversarial sequence σ.

III. Algorithms and Analysis

We consider a greedy deterministic policy $\hat{\pi}$ for the source. The source divides the whole time horizon T into equal T_1+1 blocks. Thus, each block in the partition has a length of $\frac{T}{T_1+1}$. Under the policy $\hat{\pi}$, the source directly transmits an update packet to the cache at the beginning of each block, starting from the second block and the source directly transmits an update packet to the user at the end of each block, except the last block. Finally, the source chooses any arbitrary $T_2 - T_1$ time slots from the remaining time slots and transmits to the cache. This scheme is pictorially shown in Fig. 2. Theorem 1 gives an upper bound for the competitive ratio of $\hat{\pi}$.

Theorem 1. *The policy $\hat{\pi}$ is $\left(\frac{1+T_1+T}{1+T_2+T_1+T}\right)\left(1 + \frac{T_2}{T_1+1}\right)$ competitive.*

Proof: First, note that the source can transmit to the user (directly or via the cache), only for $T_1 + T_2$ time slots. Now, we can show that if this transmission occurs with uniform spacing over time horizon T, i.e., if we divide the whole time horizon into equal length sections with the length of each section being $\frac{T}{T_1+T_2+1}$, and the age of each section evolves as $1, 2, \cdots, \frac{T}{T_1+T_2+1}$, this provides a universal lower bound on the age for the optimal policy corresponding to any adversarial action σ. Thus, for any adversarial action σ,

$$A^{o,\sigma} \geq \frac{1}{2}\left(1 + \frac{T}{T_2 + T_1 + 1}\right) \tag{3}$$

The optimal action for the adversary corresponding to the policy $\hat{\pi}$, is to never transmit any cached packet to the user, i.e., the optimal adversarial action is all zeros. For this adversarial action, the age of the user with policy $\hat{\pi}$, evolves as $1, 2, \cdots, \frac{T}{T_1+1}$ in each section. Thus,

$$A^{\hat{\pi},\sigma} = \frac{1}{2}\left(1 + \frac{T}{T_1 + 1}\right) \tag{4}$$

Thus,

$$\begin{aligned} c^{\hat{\pi}} &= \sup_{\sigma} \frac{A^{\hat{\pi},\sigma}}{A^{o,\sigma}} && (5) \\ &\leq \frac{\sup_{\sigma} A^{\hat{\pi},\sigma}}{\inf_{\sigma} A^{o,\sigma}} && (6) \\ &\leq \left(\frac{1 + T_1 + T}{1 + T_2 + T_1 + T}\right)\left(1 + \frac{T_2}{T_1 + 1}\right) && (7) \end{aligned}$$

which completes the proof. ■

Next, we find a universal lower bound on the competitive ratio for this problem. For this, we leverage Yao's minimax principle [19]. For online learning, Yao's minimax principle states that the competitive ratio for the best randomized policy,

$$\inf_{\pi \in R} c^{\pi} = \sup_{P} \inf_{\pi \in D} c^{\pi,P} \tag{8}$$

where P is the arbitrary distribution from which the adversary samples an adversarial sequence, R is the set of randomized policies. Thus, the left hand side of (8) provides the best competitive ratio possible. We denote the set of deterministic policies as D. Then, $c^{\pi,P}$ is the smallest competitive ratio for π under the distribution P, i.e., $c^{\pi,P}$ is the infimum of all c which satisfies the following,

$$\mathbb{E}_{\sigma \sim P}[A^{\pi,\sigma}] \leq c\mathbb{E}_{\sigma \sim P}[A^{o,\sigma}] + a \tag{9}$$

where a is a constant.

Now, we define a particular probability distribution P_1 to generate the adversarial sequence σ. In P_1, the adversary chooses action 0 with probability $\frac{1}{2}$ and action 1 with probability $\frac{1}{2}$. Thus, we can obtain a fundamental lower bound on the competitive ratio for this problem from Yao's minimax principle as follows. From (8),

$$\begin{aligned} \inf_{\pi \in R} c^{\pi} &\geq \inf_{\pi \in D} c^{\pi,P_1} && (10) \\ &\geq \frac{\mathbb{E}_{\sigma \sim P}[A^{\pi_D^*,\sigma}]}{\mathbb{E}_{\sigma \sim P}[A^{o,\sigma}]} && (11) \end{aligned}$$

where π_D^* is the best deterministic policy under the probability distribution P_1. Now, we find the numerator and an upper bound for the denominator of (11). In the next few lemmas, we first find the best deterministic policy for distribution P_1.

In the next lemma, we study the optimal locations for the cache updates for any arbitrary user update locations. We consider a policy $\bar{\pi}$, for which the source divides the whole time horizon into $T_2 + 1$ sections, at the end of each section the source transmits an update packet to the cache except the very last section, the length of a section in which the source transmits an update packet directly to the user is $\frac{T-T_1}{T_2+1}+1$, and the length of a section in which the source does not transmit an update packet directly to the user is $\frac{T-T_1}{T_2+1}$.

Lemma 1. *Under the constraint of our system, i.e., the source cannot transmit directly to the user in two consecutive sections, the policy $\bar{\pi}$ assigns the optimal cache update location for any arbitrary user update location.*

Proof: We consider an arbitrary ith section for the policy $\bar{\pi}$. We call the source to cache update for the $(i-1)$th section as the left cache update for the ith section and the source to cache update for the ith section as the right cache update for the ith section. First, we show that if there is a direct source to user update in the ith section, then moving the left cache update for the ith section to the left or to the right increases the expected age of the system. Then, we show that if there is no direct source to user update in the ith section, moving the left cache update for the ith section to the right or to the left by one time slot increases the expected age of the system.

We consider that there is a direct source to user update in the ith section. Now, we consider a source policy $\tilde{\pi}$, such that it is similar to the policy $\bar{\pi}$, except it shifts the left cache update for the ith section by one time slot left compared to the policy $\bar{\pi}$. Now, we show that the expected age for the policy $\tilde{\pi}$ is higher than the expected age for the policy $\bar{\pi}$. Let us assume that for the policy $\bar{\pi}$ the $(i-1)$th section ends at the time slot t_1. That means that the left cache update for the ith section occurs at the time slot t_1. We define $\frac{T-T_1}{T_2+1} = x_2$, then the ith section has x_2+1 time slots and the right cache update for the ith time slot occurs at the time slot t_1+x_2+1. Let us assume that the direct source to user update occurs at the time slot t_1+x_3. Note that, for any adversarial sequence the difference of the evolution of the age between the policies $\bar{\pi}$ and $\tilde{\pi}$ starts at the time slot t_1 and ends at an arbitrary time slot t_2, where t_2 either depends on at which time slot the source again directly transmits to the user or it depends on at which time slot the adversarial action is 1 after the ith section.

Now, we find several adversarial sequences for which the policy $\bar{\pi}$ provides lower expected age compared to the policy $\tilde{\pi}$. Similarly, we find several adversarial sequences for which the policy $\tilde{\pi}$ provides lower expected age compared to the policy $\bar{\pi}$. Finally, we show that combining the effects of all these sequences we obtain a lower expected age for the policy $\bar{\pi}$ compared to the policy $\tilde{\pi}$.

First, let us consider an adversarial sequence, σ_1, such that $\sigma_1(t_1-1)=0$, $\sigma_1(t_1)=1$, $\sigma_1(t_1+x_3+1)=1$ and $\sigma_1(t_1+x_2+1)=1$. Note that, for σ_1 the ages for the policy $\bar{\pi}$ and $\tilde{\pi}$ differ till the time slot t_1+x_2+1. Let us assume that the age of the system for any of the two policies corresponding to σ_1 at time slot t_1-1 is Δ. The evolution of the age from the time slot t_1 to the time slot t_1+x_2+1 corresponding to the policy $\bar{\pi}$ is $\Delta+1 \to 1 \to 2 \cdots \to x_3 \to 1 \to x_3+2 \to x_3+2 \to \cdots \to x_2+1$. Similarly, the evolution of the age from the from the time slot t_1 to the time slot t_1+x_2 corresponding to the policy $\tilde{\pi}$ is $\Delta+1 \to 1 \to 2 \to \cdots \to x_3+1 \to 1 \to x_3+3 \to x_3+4 \to \cdots \to x_2+2$. Thus, the difference of the ages between the policies $\tilde{\pi}$ and $\bar{\pi}$ corresponding to σ_1 is x_2. Note that, the adversary chooses the sequence σ_1 with probability $\frac{1}{2^4}$. Now, consider that at time slot $t_1+x_2+1+x_4$, the source again directly transmits an update packet to the user. Now, we assume another adversarial sequence σ_2, such that $\sigma_2(t_1-1)=0$, $\sigma_2(t_1)=1$, $\sigma_2(t_1+x_3+1)=1$, $\sigma_2(t_1+x_2+1)=0=\sigma_2(t_1+x_2+2)=\cdots=\sigma_2(t_1+x_2+k)$ and $\sigma_2(t_1+x_2+k+1)=1$, $k<x_4$. With a similar argument made for the sequence σ_1, we observe that for σ_2 the difference of the ages between $\tilde{\pi}$ and $\bar{\pi}$ is x_2+k. Note that the adversary chooses the sequence σ_2 with probability $\frac{1}{2^{4+k}}$.

From the above study it is evident that, given an adversarial sequence, σ such that $\sigma(t_1-1)=0$ and $\sigma(t_1)=1$, the difference of the expected ages between the policies $\tilde{\pi}$ and $\bar{\pi}$ increases with x_4. Let us denote the set of all adversarial sequences, σ such that $\sigma(t_1-1)=0$ and $\sigma(t_1)=1$, as $\mathcal{S}$. Now, we find the minimum difference of the ages between the policies $\bar{\pi}$ and $\tilde{\pi}$ with respect to x_4, whenever the adversary chooses a sequence from $\mathcal{S}$, i.e., when $x_4=1$. Thus, we have

$$\begin{aligned}
&\mathbb{E}[A^{\tilde{\pi},\sigma} - A^{\bar{\pi},\sigma}|\sigma \in \mathcal{S}] \\
&\quad = \sum_{i=1}^{x_2-x_3} \frac{x_2-i+1}{2^{1+i}} + \frac{2x_3}{2^{x_2-x_3+1}} + \sum_{j=0}^{x_2-x_3-2} \frac{x_2-i+1}{2^{1+i}} \qquad (12) \\
&\quad = \frac{3x_2}{2} + \frac{1}{2^{x_2-x_3+1}} - \frac{x_3}{2^{x_2-x_3+1}} - \frac{1}{2} \qquad (13)
\end{aligned}$$

The first term in (13) corresponds to the adversarial sequences in $\mathcal{S}$, for which $\sigma(t_1+x_2+1)=1$ and in any of the time slot t', from t_1+x_3+1 to t_1+x_2, $\sigma(t')=1$. The second term in (13) corresponds to the adversarial sequences in $\mathcal{S}$, for which the adversarial action is 0, from the time slot t_1+x_3+1 to the time slot t_1+x_2. The last term in (13) corresponds to the adversarial sequences in $\mathcal{S}$, for which $\sigma(t_1+x_2+1)=0$ and in any of the time slot t', from t_1+x_3+1 to t_1+x_2, such that $\sigma(t')=1$.

Now, we consider all the adversarial sequences, σ such that $\sigma(t_1-1)=1$ and $\sigma(t_1)=0$. We denote the set of all such adversarial sequences as $\mathcal{S}_1$. Following a similar analysis as we have done to obtain (13), we have the following

$$\begin{aligned}
&\mathbb{E}[A^{\bar{\pi},\sigma} - A^{\tilde{\pi},\sigma}|\sigma \in \mathcal{S}_1] \\
&\quad = \sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2-x_3} \frac{ix_2+j-2}{2^{1+i+j}} + \sum_{i=1}^{x_3-1} \frac{(i+1)x_2-x_3-1}{2^{i+x_2-x_3}} \\
&\qquad + \sum_{j=1}^{x_2-x_3} \frac{x_3x_2+j-2}{2^{x_3+j}} + \sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2-x_3} \frac{ix_2+j-3}{2^{1+i+j}} \\
&\qquad + \sum_{j=1}^{x_2-x_3} \frac{x_3x_2+j-3}{2^{x_3+j}} + \frac{x_3x_2+x_2-x_3-1}{2^{x_2-1}} \qquad (14) \\
&\quad = 2x_2 - \frac{2x_2}{2^{x_3}} - \frac{1}{2} - \frac{1}{2^{x_2-x_3+1}} \qquad (15)
\end{aligned}$$

Next, we consider all the adversarial sequences σ such that $\sigma(t_1-1)=1$ and $\sigma(t_1)=1$. We denote the set of all such adversarial sequences as $\mathcal{S}_2$. Following a similar analysis as we have done to obtain (13), we have the following

$$\mathbb{E}[A^{\bar{\pi},\sigma} - A^{\tilde{\pi},\sigma}|\sigma \in \mathcal{S}_2]$$

$$= \sum_{i=-1}^{x_2-x_3-2} \frac{i}{2^{3+i}} + \frac{(x_2-x_3-1)}{2^{x_2-x_3}} \sum_{i=-2}^{x_2-x_3-3} \frac{i}{2^{4+i}} \tag{16}$$

$$= -\frac{1}{2^{x_2-x_3+1}} - \frac{1}{2} \tag{17}$$

Finally, we consider all the adversarial sequences σ such that $\sigma(t_1-1)=0$ and $\sigma(t_1)=0$. We denote the set of all such adversarial sequences as $\mathcal{S}_3$. Following a similar analysis as we have done to obtIN (13), we have the following

$$\mathbb{E}[A^{\bar{\pi},\sigma} - A^{\bar{\pi},\sigma} | \sigma \in \mathcal{S}_3]$$

$$= \sum_{i=1}^{x_3-1} \sum_{j=1}^{x_2-x_3} \frac{x_2-i-j+1}{2^{i+j}} + \sum_{j=1}^{x_2-x_3} \frac{x_2-x_3-j+1}{2^{x_3+j-1}}$$

$$+ \sum_{i=1}^{x_3-1} \frac{x_3-i}{2^{x_2-x_3+i}} + \sum_{i=1}^{x_3-1} \sum_{j=1}^{x_2-x_3} \frac{1}{2^{i+j+1}} + \sum_{j=1}^{x_2-x_3} \frac{1}{2^{x_3+j}} \tag{18}$$

$$= x_2 + \frac{2}{2^{x_3}} + \frac{1}{2^{x_2-x_3+1}} - 2.5 \tag{19}$$

Now, combining (13), (15), (17) and (19), we obtain

$$\mathbb{E}[A^{\bar{\pi},\sigma} - A^{\bar{\pi},\sigma}] = 2 - \frac{x_2}{2} - \frac{2}{2^{x_2-x_3}} - \frac{2x_2}{2^{x_3}} + \frac{x_3}{2^{x_2-x_3+1}} - \frac{2}{2^{x_3}} \tag{20}$$

We see that (20) is an increasing function with respect to x_3. Thus, we take $x_3 = x_2$, for which we see that $\mathbb{E}[A^{\bar{\pi},\sigma} - A^{\bar{\pi},\sigma}] < 0$. In a similar fashion, we can show that, if we shift the left cache update for the ith section to the right, then the average age of the system increases. Also, in a similar fashion, we can show that, if there is no direct source to user update in the ith section, then if we move the left cache update for the ith section to the right or to the left by one time slot, then the expected age of the system increases. ■

In the next lemma, we find the optimal placement for the source to user update, with respect to the source to cache update given by Lemma 1.

Lemma 2. *Between two cache updates, i.e., the $(i-1)$th and the ith cache updates, the source achieves the minimum age by updating the user directly just before the ith cache update.*

Proof: Consider the source policy given in Lemma 1, i.e., policy $\bar{\pi}$. Consider that at the ith section, the source transmits an update packet directly to the user. Let us assume that the ith section starts at t_1, the source to user update occurs at t_1+x_3 and the age of the user at time slot t_1+x_3 is Δ. Now, consider another policy $\tilde{\pi}$ which is similar to $\bar{\pi}$, except it transmits the direct source to user update at the ith section at time slot t_1+x_3+1, instead of time slot $t+x_3$. Now, we show that under the distribution P_1, the policy $\tilde{\pi}$ results in lower expected age for the user compared to policy $\bar{\pi}$.

Following a similar analysis as of the proof of Lemma 1,

$$\mathbb{E}[A^{\tilde{\pi},\sigma} - A^{\bar{\pi},\sigma}] \leq 0 \tag{21}$$

which concludes the proof of this lemma. ■

From Lemma 1 and Lemma 2, we see that the optimal action for the source is to divide the whole time horizon into T_2+1 sections, and transmit an update packet to the cache at the end of each section except for the last section. The length of a section in which there is no user update is $\frac{T-T_1}{T_2+1}$ and the length of a section in which there is a user update is $\frac{T-T_1}{T_2+1}+1$. The optimal source to user update locations are just before the source to cache updates. From the symmetry of the cache updates over the time horizon, we claim that the optimal source to user update locations are equidistant over the time horizon T, that is there are $\frac{T_2}{T_1}$ source to cache updates between two consecutive source to user updates. Let us denote this source policy as $\bar{\pi}$. According to (11), to get a lower bound on the competitive ratio, we first find the average age for the policy $\bar{\pi}$; see Fig. 3 for a pictorial representation of the policy $\bar{\pi}$.

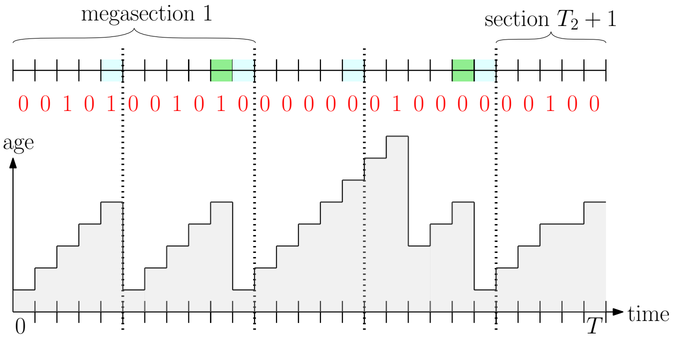

Fig. 3. Optimal deterministic algorithm, dividing timelines into T_2+1 sections of equal size. $T_1=2$, $T_2=4$, $T=27$.

A. Average Age for Policy $\bar{\pi}$

For the notational convenience, in this section, we denote the source to cache update as the cache update and the source to user update as the user update. Recall that for the policy $\bar{\pi}$ a section i ends with the ith cache update and starts right after the $(i-1)$th cache update. We define a megasection which consists of $\frac{T_2}{T_1}$ slots. The ith mega section ends with a source update followed by a cache update, and starts right after the $(i-1)$th megasection. Recall that, for policy $\bar{\pi}$, each section which does not consist of a user update, has x_2 time slots, where $x_2 = \frac{T-T_1}{T_2+1}$ and the sections which consist of a user update have x_2+1 time slots. Recall that, for a policy π and an adversarial sequence σ, the age of the user at the tth time slot is defined as $v^{\pi,\sigma}(t)$. For notational convenience, in any arbitrary section, we redefine the age of the user at the jth time slot of that section as v_j, where either $j \in 1,2,\cdots,x_2$ or $j \in 1,2,\cdots,x_2+1$, depending on the section. We define v_0 to be the age of the user at the last time slot of the preceding section (indicated by time slot 0). If a user does not receive a direct update from the source in time slot $j-1$, then we have

$$\mathbb{E}[v_j | v_{j-1}] = \frac{(v_{j-1}+1)}{2} + \frac{j}{2} \tag{22}$$

$$= \frac{v_{j-1}}{2} + \frac{(j+1)}{2} \tag{23}$$

Note that, $\mathbb{E}[v_j | v_{j-1}, v_0] = \mathbb{E}[v_j | v_{j-1}]$ as v_j is completely determined by v_{j-1} and the action taken by the adversary in time slot $j-1$, which gives

$$\mathbb{E}[v_j|v_0] = \frac{\mathbb{E}[v_{j-1}|v_0]}{2} + \frac{(j+1)}{2} \tag{24}$$

Iteratively repeating this for $\mathbb{E}[v_{j-1}|v_0]$, we obtain

$$\mathbb{E}[v_j|v_0] = \frac{v_0}{2^j} + \sum_{k=2}^{j+1} \frac{k}{2^{j+2-k}} \tag{25}$$

Note that the user updates occur only in time slot x_2 of the last section of a megasection, hence (25) is always applicable to time slots $\{1, \ldots, x_2\}$ of every section.

Conditioned on v_0, the sum of age of time slots $\{0, \ldots, x_2-1\}$ of this section, is

$$\begin{aligned}
\mathbb{E}\Bigg[\sum_{j=0}^{x_2-1} v_j \Bigg| v_0\Bigg] &= \sum_{j=0}^{x_2-1} \left(\frac{v_0}{2^j} + \sum_{k=2}^{j+1} \frac{k}{2^{j+2-k}}\right) &(26)\\
&= \sum_{j=0}^{x_2-1} \frac{v_0}{2^j} + \sum_{j=1}^{x_2-1} \sum_{k=2}^{j+1} \frac{k}{2^{j+2-k}} &(27)\\
&= \sum_{j=0}^{x_2-1} \frac{v_0}{2^j} + \sum_{j=1}^{x_2-1} \frac{1}{2^{j+2}} \sum_{k=2}^{j+1} 2^k k &(28)\\
&= v_0\alpha + \beta &(29)
\end{aligned}$$

where α is computed as

$$\alpha = \sum_{j=0}^{x_2-1} \frac{1}{2^j} = \frac{1-\frac{1}{2^{x_2}}}{1-0.5} = 2\left(1 - \frac{1}{2^{x_2}}\right) \tag{30}$$

and β is computed as

$$\begin{aligned}
\beta &= \sum_{j=1}^{x_2-1} \frac{1}{2^{j+2}} \sum_{k=2}^{j+1} 2^k k &(31)\\
&= \sum_{j=1}^{x_2-1} \frac{1}{2^{j+2}} (2\frac{1-(j+2)2^{j+1}+(j+1)2^{j+2}}{(1-2)^2} - 2) &(32)\\
&= \frac{1}{(1-2)^2}\left(\sum_{j=1}^{x_2-1} \frac{1}{2^{j+1}} + \sum_{j=1}^{x_2-1} j\right) - \sum_{j=1}^{x_2-1} \frac{1}{2^{j+1}} &(33)\\
&= \sum_{j=1}^{x_2-1} j = \frac{(x_2-1)x_2}{2} &(34)
\end{aligned}$$

Using nested expectations, we obtain

$$\mathbb{E}\Bigg[\sum_{j=0}^{x_2-1} v_j\Bigg] = \mathbb{E}[v_0]\alpha + \beta \tag{35}$$

Let us assume that $\frac{T_2}{T_1} = x_3$. Next, instead of considering the age of arbitrary sections, let the sections of a megasection be indexed by i, where $i \in \{1, \ldots, x_3\}$ and let $v_{j,i}$ denote the age of time slot i of section j of arbitrary megasection. Building on (29), let $v_{0,i}$ denote the age of last time slot of the section preceding section i, i.e., $v_{x_2,i-1} = v_{0,i}$ for $i > 1$. Note that $v_{0,1} = v_{x_2+1,x_3} = 1$, since a user update occurs in the second last time slot of every megasection, causing the age in the last time slot of a megasection to drop to 1. Consequently, the sum of expected age of all time slots in a megasection is

$$\begin{aligned}
\mathbb{E}\Bigg[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\Bigg] &= \mathbb{E}\Bigg[\sum_{i=1}^{x_3}\sum_{j=0}^{x_2-1} v_{j,i} + v_{x_2,x_3}\Bigg] &(36)\\
&= \mathbb{E}\Bigg[\sum_{i=1}^{x_3} v_{0,i}\Bigg]\alpha + \beta_1 &(37)
\end{aligned}$$

with $\beta_1 = \mathbb{E}\big[v_{x_2,x_3}\big] + x_3\beta$, where $\mathbb{E}\big[v_{x_2,x_3}\big]$ can be obtained from (25).

Given $v_{0,1} = 1$, we first compute the expected value of $v_{0,i} = v_{x_2,i-1}$ in terms of $v_{0,i-1}$ using (25) by substituting $j = x_2$,

$$\begin{aligned}
\mathbb{E}[v_{0,i}|v_{0,i-1}] &= \frac{v_{0,i-1}}{2^{x_2}} + \sum_{k=2}^{x_2+1} \frac{k}{2^{x_2+2-k}} &(38)\\
&= rv_{0,i-1} + s &(39)
\end{aligned}$$

where r is computed as

$$r = \frac{1}{2^{x_2}} \tag{40}$$

and s is computed as

$$\begin{aligned}
s &= \sum_{k=2}^{x_2+1} \frac{k}{2^{x_2+2-k}} &(41)\\
&= \frac{1}{2^{x_2+2}} \sum_{k=2}^{x_2+1} 2^k k &(42)\\
&= x_2 &(43)
\end{aligned}$$

Iteratively repeating this for all $v_{0,\ell}$, we obtain

$$\mathbb{E}[v_{0,i}] = r^{i-1}v_{0,1} + \sum_{\ell=0}^{i-2} r^\ell s \tag{44}$$

where $v_{0,1} = 1$. Combining (37) and (44), we have

$$\begin{aligned}
&\mathbb{E}\Bigg[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\Bigg]\\
&=\Bigg[\sum_{i=1}^{x_3} r^{i-1} + s\sum_{i=2}^{x_3}\sum_{\ell=0}^{i-2} r^\ell\Bigg]\alpha + \beta_1 &(45)\\
&=\frac{x_3x_2(x_2-1)}{2} + 2x_2x_3 + 2 - \frac{1}{2^{x_2x_3}}\\
&\quad + \frac{x_2 2^{x_2}}{2^{x_2}-1}\left(\frac{1}{2^{x_2x_3}} - 1\right) &(46)
\end{aligned}$$

Let us define the total age of the ith megasection as A_i, $i \in \{1, 2, \cdots, x_3\}$. Note that, the random variables A_i are independent and identically distributed, thus the average expected age of the user for the policy $\check{\pi}$ and the adversarial sequence following a uniform probability distribution is,

$$\mathbb{E}_{\sigma\sim P_1}[A^{\check{\pi},\sigma}] = \frac{1}{T}\sum_{i=1}^{T_1} A_i = \frac{T_1}{T}\frac{1}{T_1}\sum_{i=1}^{T_1} A_i \tag{47}$$

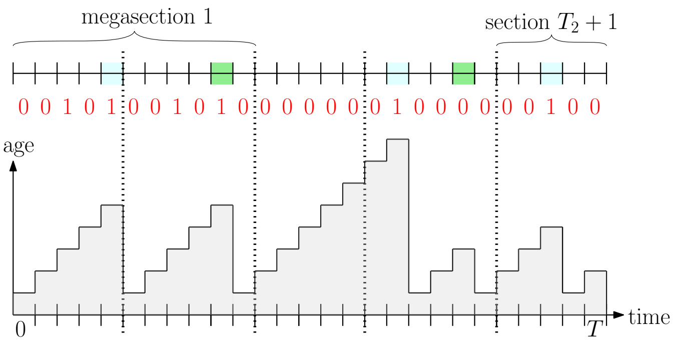

Fig. 4. A policy $\breve{\pi}_1$ to find an upper bound on the offline optimal policy, dividing timelines into T_2+1 sections of equal size. $T_1=2$, $T_2=4$, $T=27$.

From (47), for large T_1, as $\frac{T}{T_1}$ is constant, i.e., for large T,

$$\mathbb{E}_{\sigma\sim P_1}[A^{\breve{\pi},\sigma}] = \frac{T_1}{T}\mathbb{E}\left[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\right] \tag{48}$$

B. Upper Bound on the Offline Optimal Policy

Next, we find an upper bound on the expected age of the offline optimal policy, where the adversarial sequences follow the probability distribution P_1. Let us define a policy $\breve{\pi}_1$ which is similar to the policy $\breve{\pi}$, except for policy $\breve{\pi}_1$ the source to cache update does not occur at the end of each section but a cache update can occur at any time slot between the end of a section and the end of the next section depending on whenever the adversarial sequence is 1 for the first time. A pictorial representation of this policy is shown in Fig. 4.

As the structure of both the policies $\breve{\pi}$ and $\breve{\pi}_1$ are the same, we again define the megasection and the section. As before, the number of sections in a megasection is x_3, the number of time slots in a section in which there is no direct source to user update is x_2 and the other sections have x_2+1 time slots. Then, we have

$$\mathbb{E}[v_j|v_0] = \frac{v_0+j}{2^j} + \sum_{k=0}^{j-1}\frac{j-k}{2^{k+1}} \tag{49}$$

This is because, if the adversary does not transmit any packet in the time slots $\{0,\dots,j-1\}$, which has the probability $\frac{1}{2^j}$, the user age increments by j units over v_0 in time slot j. However, if for some $k<j$, the adversary forwards a packet to the user for the first time in time slot k, which has probability $\frac{1}{2^{k+1}}$, the user age at time slot j would be $j-k$.

As before, this formula is applicable in absence of user update before timeslot j in that section, which occurs only at the second last timeslot of the last section of a megasection, hence this formula is always applicable for $j=\{1,\dots,x_2\}$.

Conditioned on a_0, the sum of age of timeslots $\{0,\dots,x_2-1\}$ of this section is

$$\mathbb{E}\left[\sum_{j=0}^{x_2-1} v_j \middle| v_0\right] = \sum_{j=0}^{x_2-1}\left(\frac{v_0+j}{2^j} + \sum_{k=0}^{j-1}\frac{j-k}{2^{k+1}}\right) \tag{50}$$

$$= v_0\bar{\alpha}+\bar{\beta} \tag{51}$$

where $\bar{\alpha}$ is

$$\bar{\alpha} = 2\left(1-\frac{1}{2^{x_2}}\right) \tag{52}$$

and $\bar{\beta}$ is

$$\bar{\beta} = 4 - \frac{2x_2}{2^{x_2}} - \frac{4}{2^{x_2}} + \frac{(x_2-3)x_2}{2} \tag{53}$$

Approaching as in (37), let $a_{j,i}$ denote the age of the jth timeslot of the ith section of a megasection, such that $a_{0,1}=1=a_{x_2+1,x_3}$. Then, the sum of expected age of all timeslots in a megasection is

$$\mathbb{E}\left[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\right] = \mathbb{E}\left[\sum_{i=1}^{x_3}\sum_{j=0}^{x_2-1} v_{j,i} + v_{x_2,x_3}\right] \tag{54}$$

$$= \mathbb{E}\left[\sum_{i=1}^{x_3} v_{0,i}\right]\bar{\alpha}+\bar{\beta}_1 \tag{55}$$

with $\bar{\beta}_1 = \mathbb{E}[v_{x_2,x_3}] + x_3\bar{\beta}$, where v_{x_2,x_3} is obtained from (49). Given $v_{0,1}=1$, we compute the expected value of $v_{0,i}=v_{x_2,i-1}$ in terms of $v_{0,i-1}$ using (49) by substituting $j=x_2$

$$\mathbb{E}[v_{0,i}|v_{0,i-1}] = \frac{v_{0,i-1}+x_2}{2^{x_2}} + \sum_{k=0}^{x_2-1}\frac{x_2-k}{2^{k+1}} \tag{56}$$

$$= \bar{r}v_{0,i-1}+\bar{s} \tag{57}$$

where $\bar{r}$ is

$$\bar{r} = \frac{1}{2^{x_2}} \tag{58}$$

and $\bar{s}$ is

$$\bar{s} = \frac{x_2}{2^{x_2}} + x_2 + \frac{1}{2^{x_2}} - 1 \tag{59}$$

Iteratively repeating this for all $v_{0,\ell}$, we obtain

$$\mathbb{E}[v_{0,i}] = \bar{r}^{i-1}v_{0,1} + \sum_{\ell=0}^{i-2}\bar{r}^{\ell}\bar{s} \tag{60}$$

where $v_{0,1}=1$. Summing over all i, (54) becomes

$$\mathbb{E}\left[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\right]$$

$$= \left[\sum_{i=1}^{x_3}\bar{r}^{i-1} + \bar{s}\sum_{i=2}^{x_3}\sum_{\ell=0}^{i-2} r^{\ell}\right]\bar{\alpha}+\bar{\beta}_1 \tag{61}$$

$$= 3 - \frac{2x_3}{2^{x_2}} + 2x_3 + x_2 - \frac{2}{2^{x_2x_3}} + \frac{x_3x_2(x_2-3)}{2} + 2x_3x_2 + \frac{x_2(2^{x_2}+1)}{2^{x_2x_3}(2^{x_2}-1)} - \frac{2x_2 2^{x_2}}{2^{x_2}-1} \tag{62}$$

Similar to (48), we obtain

$$\mathbb{E}_{\sigma\sim P_1}[A^{o,\sigma}] \le \frac{T_1}{T}\mathbb{E}\left[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\right] \tag{63}$$

Thus, from (11), (48) and (63), we obtain a universal lower bound for the competitive ratio for this system model.

IV. Conclusion

In this paper, we considered a system where a source aims to minimize the age of a user by transmitting fresh update packets to the user over a time horizon T time slots. The source can directly transmit update packets to the user for T_1 time slots. The source can also transmit update packets to a cache in the system for T_2 time slots, where $T_1 \ll T_2$ and $T_1 + T_2 < T$. There is an adversary in the system which completely controls the cache to user communication link. Whenever the user receives a cached update packet, the user compares the time stamp of the packet it has with the time stamp of the received packet, and keeps only the fresher packet while discards the staler packet. The adversary can change the time stamp of the cached update packet, and by doing so, it can deceive the user by making it store a staler update packet with the cost of discarding a fresher update packet. The goal of the adversary is to increase the age of the user, while the goal of the source is to minimize the age of the user. We formulated this problem as an online learning problem and studied the competitive ratio for this problem. First, we proposed a deterministic algorithm and provided an upper bound on the competitive ratio for the proposed algorithm. Then, we proposed a universal lower bound for the studied system model. The extensions of the model with multiple users and/or multiple caches are interesting future directions.

References

[1] R. Bhattacharjee, S. Banerjee, and A. Sinha. Fundamental limits on the regret of online network-caching. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2):1–31, June 2020.

[2] A. Kosta, N. Pappas, and V. Angelakis. Age of information: A new concept, metric, and tool. *Foundations and Trends in Networking*, 12(3):162–259, November 2017.

[3] Y. Sun, I. Kadota, R. Talak, and E. Modiano. Age of information: A new metric for information freshness. *Synthesis Lectures on Communication Networks*, 12(2):1–224, December 2019.

[4] R. D. Yates, Y. Sun, R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus. Age of information: An introduction and survey. *IEEE Journal on Selected Areas in Communications*, 39(5):1183–1210, May 2021.

[5] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger. Age-optimal constrained cache updating. In *IEEE ISIT*, June 2017.

[6] A. Garnaev, W. Zhang, J. Zhong, and R. D. Yates. Maintaining information freshness under jamming. In *IEEE Infocom*, May 2019.

[7] G. D. Nguyen, S. Kompella, C. Kam, J. E. Wieselthier, and A. Ephremides. Impact of hostile interference on information freshness: A game approach. In *IEEE WiOpt*, May 2017.

[8] Y. Xiao and Y. Sun. A dynamic jamming game for real-time status updates. In *IEEE Infocom*, April 2018.

[9] S. Banerjee, R. Bhattacharjee, and A. Sinha. Fundamental limits of age-of-information in stationary and non-stationary environments. In *IEEE ISIT*, June 2020.

[10] R. Bhattacharjee and A. Sinha. Competitive algorithms for minimizing the maximum age-of-information. In *ACM Perf. Eval.*, September 2020.

[11] S. Banerjee and S. Ulukus. Age of information in the presence of an adversary. In *IEEE Infocom*, May 2022.

[12] S. Banerjee and S. Ulukus. Game theoretic analysis of an adversarial status updating system. In *IEEE ISIT*, June 2022.

[13] S. Banerjee and S. Ulukus. The freshness game: Timely communications in the presence of an adversary. *Available at arXiv:2302.14024*.

[14] S. Banerjee, S. Ulukus, and A. Ephremides. Age of information of a power constrained scheduler in the presence of a power constrained adversary. In *IEEE Infocom*, May 2023.

[15] P. Kaswan and S. Ulukus. Age of gossip in ring networks in the presence of jamming attacks. In *Asilomar Conference*, October 2022.

[16] P. Kaswan and S. Ulukus. How robust are timely gossip networks to jamming attacks? Available at arXiv:2303.11310.

[17] P. Kaswan and S. Ulukus. Susceptibility of age of gossip to timestomping. In *IEEE ITW*, November 2022.

[18] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge university press, 1995.

[19] S. Albers. *Competitive Online Algorithms*. Citeseer, 1996.