

Information Freshness in Multihop Wireless Networks

Vishrant Tripathi^{id}, Graduate Student Member, IEEE, Rajat Talak^{id}, and Eytan Modiano, Fellow, IEEE

Abstract—We consider the problem of minimizing age of information in multihop wireless networks and propose three classes of policies to solve the problem - stationary randomized, age difference, and age debt. For the unicast setting with fixed routes between each source-destination pair, we first develop a procedure to find age optimal *Stationary Randomized* policies. These policies are easy to implement and allow us to derive closed-form expression for average AoI. Next, for the same unicast setting, we develop a class of heuristic policies, called *Age Difference*, based on the idea that if neighboring nodes try to reduce their age differential then all nodes will have fresher updates. This approach is useful in practice since it relies only on the local age differential between nodes to make scheduling decisions. Finally, we propose the class of policies called *Age Debt*, which can handle 1) non-linear AoI cost functions; 2) unicast, multicast and broadcast flows; and 3) no fixed routes specified per flow beforehand. Here, we convert AoI optimization problems into equivalent network stability problems and use Lyapunov drift to find scheduling and routing schemes that stabilize the network. We also provide numerical results comparing our proposed classes of policies with the best known scheduling and routing schemes available in the literature for a wide variety of network settings.

Index Terms—Age of information, wireless networks, scheduling, multi-hop networks.

I. INTRODUCTION

EMERGING applications such as networked control systems, real-time surveillance and monitoring, augmented and virtual reality, cloud gaming, and caching at the wireless edge rely crucially on the continuous delivery of fresh updates over communication networks. Further, exchanging fresh information updates over *multi-hop* wireless networks is gaining increasing relevance with the advent of ad-hoc networked wireless systems such as internet of things (IoT), vehicular networks, and networks of unmanned aerial vehicles.

These systems differ from the traditional communication systems in two ways. In traditional communication systems, data or packet arrival is assumed to be an exogenous process that cannot be controlled. However, in a lot of real-time

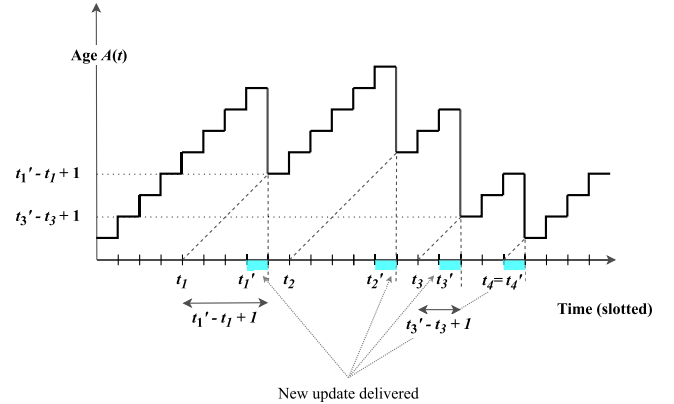


Fig. 1. Age of Information (AoI) as a function of time. Here, t_i is the time of generation of the i th packet at the source, and t'_i is the time of its reception at the destination node.

applications, the generation of update packets, such as sensor data, can be controlled. It has been shown [1] that generating update packets at the right rate can improve freshness, striking a balance between too high a rate of generation that results in network congestion and too low a rate that results in updates being sent too infrequently.

Secondly, traditional communication systems use packet centric performance measures such as throughput or delay to characterize performance. These performance measures do not fully capture the information freshness paradigm. For example, delay of a stale packet, that got caught in the network due to network clogging, doesn't need to be accounted for as long as the intended ground station gets fresh information regularly via other, promptly received, update packets.

A new performance measure, called Age of information (AoI), was proposed in [2] and [1] to measure information freshness at the destination node. AoI at the destination node at time t , is the time elapsed since the last received update packet was generated. Figure 1, plots AoI evolving in time. Whenever the destination node receives a fresh update packet, the AoI drops to the time elapsed since the received packet's generation time, while it grows linearly otherwise.

Over the past decade, there has been a rapidly growing body of work analyzing AoI in different queuing models [1], [3], [4], [5], [6], [7], [8], [9], [10] and as a scheduling metric in single-hop wireless networks [11], [12], [13], [14], [15], [16], [17]. Next, we review these works briefly. For detailed surveys on AoI literature, we point the reader to [18] and [19].

Age for FIFO M/M/1, M/D/1, and D/M/1 queues was analyzed in [1], multiclass FIFO M/G/1 and G/G/1 queues

Manuscript received 17 November 2021; revised 27 June 2022; accepted 16 August 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Moharir. Date of publication 30 August 2022; date of current version 18 April 2023. This work was supported in part by the NSF under Grant AST-1547331, Grant CNS-1713725, and Grant CNS-1701964; and in part by the Army Research Office (ARO) under Grant W911NF17-1-0508. (Corresponding author: Vishrant Tripathi.)

The authors are with the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 USA (e-mail: vishrant@mit.edu; talak@mit.edu; modiano@mit.edu).

Digital Object Identifier 10.1109/TNET.2022.3201751

1558-2566 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

were studied in [3], while last-in-first-out (LIFO) queues under various arrival and service time distributions were studied in [4], [5], and [6]. AoI for M/M/2 and M/M/ ∞ queues was analyzed in [7] and [8], which primarily studied the impact of out-of-order delivery of packets on age. Effects of packet error or packet drop on age for the M/M/1 queue, with FIFO service, was studied in [9]. Closed-form expressions for the stationary distribution of AoI in single-server queues were derived in [10].

More recently, a number of works have also looked at AoI as a metric for designing wireless scheduling policies and solving the problem of minimizing AoI in *single-hop* wireless networks. In [11], [12], [13], [14], [15], [16], and [17], the authors consider the problem of minimizing AoI in multiple access type networks with nodes and a single base station, where only a few links can be activated at any given time. These works typically prove constant factor optimality of three classes of policies - randomized, max-weight and Whittle index based; under both reliable and unreliable channels. Slotted ALOHA-like random access for AoI minimization has also been studied in multiple recent works [20], [21], [22]. Further, minimizing general non-linear cost functions of AoI in single-hop wireless networks has been considered in [23] and [24].

AoI in *multi-hop* networks has received limited attention. AoI in multi-hop networks of queues was studied in [25], where LIFO queue service was shown to reduce age. Scaling of AoI in multihop multicast networks was studied in [26]. The optimization of routing paths for a single source sending updates has been considered in [27]. These works analyze AoI for specific network settings and do not consider scheduling policy design or handling of multiple sources, different flow-types or general AoI cost functions.

The main focus of our work is designing scheduling and routing policies to minimize AoI cost functions over general multi-hop wireless networks, while handling multiple sources; unicast, multicast and broadcast flows; as well as unreliable links. Closer to our work, for the specific case of multihop wireless networks with only all-to-all *broadcast flows*, AoI minimization has been considered in [28] and [29].

We observe that finding low complexity near optimal scheduling and routing schemes for AoI minimization which handle general network topologies, interference constraints, cost functions, different types of flows and link reliabilities has remained an open problem.

A. Contributions

In this work, we develop a unifying framework for making routing and scheduling decisions that minimize AoI cost in general multihop networks. In Section II, we describe the system model for multihop networks with general interference constraints; unicast, multicast and broadcast flows; general non-linear cost functions of AoI; and unreliable links. We consider the problem of minimizing long-term AoI cost over such networks.

In Section III, we consider the simple class of stationary randomized policies, where scheduling and routing decisions are

taken in an i.i.d. manner from a fixed probability distribution. We restrict analysis of this class of policies to weighted-sum AoI minimization over multihop networks with *only* unicast flows and *known, fixed* paths between each source-destination pair, i.e. we only need to make scheduling decisions and not routing decisions. First, we derive a closed form expression for the average AoI for each source-destination pair under any specified randomized policy. We then show that finding the best stationary randomized policy for AoI minimization over multihop networks can be converted into an equivalent single-hop AoI minimization problem. We discuss examples of how to solve this optimization problem and provide performance bounds which suggest that even the best randomized policies can be far from optimal in large networks.

In Section IV, we develop a heuristic policy called the *Age Difference* policy, based on the idea that if the age differential between nodes is small, all nodes can get fresh updates and have low AoI. We also restrict the discussion of this policy to weighted-sum AoI minimization over multihop networks with *only* unicast flows and *known, fixed* paths between each source-destination pair. We show that the age difference policy is a myopic policy that greedily optimizes for a specific AoI cost in every time-slot. We further discuss simple examples that illustrate how the age difference policy outperforms all stationary randomized policies.

In Section V, we consider the multihop problem in full generality - 1) with non-linear AoI cost functions; 2) unicast, multicast and broadcast flows and 3) considering both scheduling and routing decisions for optimization. We provide a recipe to transform AoI optimization problems into network stability problems. Instead of trying to solve for the best scheduling and routing policies directly, we assume that we have access to a set of target values which represent the average age cost for every flow in the network. These target values could be application specific freshness requirements provided by a network administrator, or they could be the solution to an optimization program that optimizes some utility function of the average age costs.

In Section V-A, we introduce the notion of *Age Debt* and set up a virtual queuing network that is stable if and only if there exists a feasible network control policy that can achieve the specified target costs. In Section V-B, we use Lyapunov drift based methods to stabilize this system of virtual queues and achieve the desired target age costs. In Section V-D, we further discuss how to choose the right age cost targets, when there is no access to either an optimization oracle or a system administrator specifying requirements for each flow. Finally, in Section VI, we provide detailed simulation results that compare our proposed AoI optimization methods with prior works. We find that Age Debt and its variants perform as well as or better than the best known scheduling and routing schemes in a wide variety of network settings.

B. Prior Versions

Preliminary versions of this work appeared in Allerton 2017 [30] and in the INFOCOM AoI Workshop 2021 [31]. In [30], we introduced stationary randomized policies for

multihop networks while in [31], we introduced the Age Debt policy. This paper combines these two lines of inquiry into a general framework for AoI optimization over multihop wireless networks. In addition, we also propose and analyze a third policy for AoI minimization called *Age Difference* and provide more substantial numerical results.

II. SYSTEM MODEL

Consider a network with N nodes connected by a fixed undirected graph $G(V, E)$. An edge (i, j) means that nodes i and j can send packets to one another directly. We assume that at most one update can be sent over an edge in any given time-slot and takes exactly one time-slot to get delivered. We normalize the time-slot duration to unity.

Flows. The network consists of K ($\leq N$) source nodes that generate information updates. All the sources are active, i.e. they generate fresh updates on demand. A source node k has to send these updates to a set of $D_k \subset N$ destination nodes in the network. We assume that a set of nodes $C_k \subset N$ is commissioned for each flow k to forward its update packets to destination nodes. For example, a network administrator could restrict the paths over which certain flows are allowed. A flow is characterized by the triplet of source node, its commissioned nodes, and the destination nodes, namely (k, C_k, D_k) . For simplicity, we use k to denote both the source node k and the flow corresponding to source node k . Note that a node could be both a destination node and also a commissioned node forwarding packets for flow k , i.e. $C_k \cap D_k$ is not necessarily empty.

Flows can be of three types depending on the number of destination nodes: (1) unicast: the flow has a single destination node. (2) multicast: the flow has multiple destination nodes, which are a strict subset of the remaining nodes. (3) broadcast: every node other than the source itself is a destination node. The commissioned nodes can be either a small subset of nodes in the network needed to reach all the destination nodes, or the entire network. We assume there to be no queuing at any node and that each node maintains a single packet buffer for the freshest packet of each flow.¹

Interference and Link States. We consider unreliable links as well as general interference constraints, i.e., transmission on all the links cannot happen simultaneously. We enumerate the set of all possible interference free choices of links and corresponding flow transmissions in the set $\mathcal{A}$. Thus, a member of set $\mathcal{A}$ contains a subset of links and corresponding flows which can be sent on these links in a single time-slot without interference. A valid network control policy must choose an action that is a member of the set $\mathcal{A}$ in every time-slot. Note that this description of $\mathcal{A}$ is very general and allows for interference constraints that depend on flow assignments. For example, consider a setting where a node is allowed to broadcast updates of a single flow to all of its neighbors in a single timestep but not send updates regarding *different* flows to each neighbor simultaneously.

¹Discarding older packets, or equivalently, preemptive LCFS (last come first serve) is known to be the optimal queuing discipline for AoI minimization [32].

For link $(i, j) \in E$, we use $U_{ij}^k(t)$ and $S_{ij}(t)$ (both $\in \{0, 1\}$) to denote the transmission decision and link state of the link (i, j) at time t . $U_{ij}^k(t)$ is 1 if a transmission of a flow- k update is scheduled on the link, at time t , and is 0 otherwise. Whereas, $S_{ij}(t)$ is 1 if a scheduled transmission on the link, at time t , will succeed; provided there is no interference. We assume $\{S_{ij}(t)\}_{t,(i,j)}$ to be independent and identically distributed processes across time t and links (i, j) , with $\gamma_{ij} = \mathbb{P}[S_{ij}(t) = 1]$.

Age Evolution. For a flow k , each commissioned and destination node keeps track of the age of the freshest packet it has received. For a node $j \in C_k \cup D_k$, we denote its age for the k th flow by $A_j^k(t)$ and it evolves as:

$$A_j^k(t+1) = \begin{cases} \min(A_j^k(t), A_i^k(t)) + 1 & \text{if } U_{ij}^k(t)S_{ij}(t) = 1 \\ A_j^k(t) + 1, & \text{if } U_{ij}^k(t)S_{ij}(t) = 0, \end{cases} \quad (1)$$

for all $i \in \{k\} \cup C_k$, $j \in C_k \cup D_k$, and link $(i, j) \in E$. Note that for any flow, the source node and the commissioned nodes transmit the update packets, while other commissioned nodes and the destination nodes receive them.

Information Freshness. We consider two metrics of information freshness. The first is the average weighted sum AoI at the destination nodes:

$$A_{\text{ave}} = \lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T \sum_{k=1}^K \sum_{j \in D_k} w_j^k A_j^k(t) \right], \quad (2)$$

where w_j^k denote constant weights, which determines the relative importance of a destination j and flow k , with respect to others. For the second metric, we consider general possibly non-linear functions of age. We associate a monotone increasing age cost function for each source-destination pair (k, j) , where $j \in D_k$, denoted by $g_j^k(\cdot)$. We define the non-linear, effective age process to be:

$$B_j^k(t) \triangleq g_j^k(A_j^k(t)), \quad (3)$$

for all $t \geq 1$. The non-linear age metric is defined as:

$$B_{\text{ave}} = \lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T \sum_{k=1}^K \sum_{j \in D_k} B_j^k(t) \right], \quad (4)$$

which is a generalized version of A_{ave} in (2).

Our goal is to minimize either A_{ave} or B_{ave} for a general, multi-hop network, by determining a policy that controls the link transmissions. A control policy needs to specify not only which links should be scheduled in each time-slot but also which flows should be transmitted along each link. We assume a centralized controller.

In the next three sections, we propose three different policies, in the order of their complexity and performance, to minimize the information freshness metrics.

III. STATIONARY RANDOMIZED POLICY

In this section, we look at the simplest case of our general multi-hop model. We consider a class of policies called stationary randomized policies, in which each feasible action $a \in \mathcal{A}$

is activated with a given probability, independently across time. The analysis of these policies is limited to the setting where each flow is *unicast* and consists of a *single known path* to forward updates from the source to each destination. We derive the exact expression of the average age metric in this case and show how to optimize it over the interference constraints.

A. Stationary Randomized Policies

We first define the space of stationary randomized policies over a multi-hop network. Let $\mathcal{E}_{(i,j)}^k(t)$ be the event that a transmission over link (i, j) is attempted, to send updates of flow k , at time-slot t .

Definition π is a *stationary randomized policy* if the events $\mathcal{E}_{(i,j)}^k(t)$ and $\mathcal{E}_{(i,j)}^k(t')$ are independent and stationary for any $t \neq t'$, for all links (i, j) and flows k . Further

$$\mathbb{P}[\mathcal{E}_{(i,j)}^k(t)] = \mathbb{P}[\mathcal{E}_{(i,j)}^k(t')] = f_{ij}^k,$$

for all time-slots t and t' . Here f_{ij}^k is the frequency of occurrence of the event $\mathcal{E}_{(i,j)}^k$.

The definition above is standard for the class of stationary randomized policies in networking literature. However, works analyzing Markov decision processes (MDPs) often use a different way to define randomized policies - where actions are sampled from a distribution that can also depend on the current state. In this section, we will focus on stationary randomized policies where actions are sampled independent of network state.

Note that all stationary randomized policies are associated with probabilities f_{ij}^k . We refer to f_{ij}^k as the *link activation frequency* of link (i, j) for flow k , and use $\mathbf{f}^k$ to denote the tuple $\{f_{ij}^k\}_{(i,j) \in E}$. A way to generate the space of all stationary randomized policies possible with a centralized scheduler is the following: transmit on all links the corresponding flow choices for an action $m \in \mathcal{A}$, with probability x_m , independently across time-slots t . The probabilities x_m can be varied to produce different stationary randomized policies, but are naturally constrained by the fact that they must sum to unity, namely, $\sum_{m \in \mathcal{A}} x_m = 1$. We use $\mathbb{X}$ to denote the space of $\mathbf{x} = (x_m)_{m \in \mathcal{A}}$ such that x_m s sum to unity.

This induces the link activation frequency f_{ij}^k given by

$$f_{ij}^k = \sum_{\{m \in \mathcal{A}: (U_{ij}^k=1) \in m\}} x_m, \quad (5)$$

for all links (i, j) and flows k . The above equation simply states that the link activation frequency f_{ij}^k is the sum of the activation probabilities of all actions $m \in \mathcal{A}$ in which a flow k packet is transmitted over link (i, j) , i.e. $U_{ij}^k = 1$. We use $\mathbf{f} = M\mathbf{x}$ to denote (5) and $\mathcal{F}$ to denote the space of all feasible link activation frequencies $\{\mathbf{f}^k\}_k = \mathbf{f}$. We will see that this space plays a critical role in determining the stationary randomized policy that minimizes average age.

B. General Network

We now consider the average age minimization problem with K flows. We assume a single destination node for every

flow. Each source node k is assigned a set of nodes C_k in the network that forms a single connected path from the source to the destination node d_k . We use p^k to denote the set of all links on the source-destination path induced by nodes $\{k\} \cup C_k \cup \{d_k\}$. The goal is to determine the optimal stationary randomized scheduling policy that minimizes the weighted average age in (2). For simplicity, we denote the weight for each source-destination pair (k, d_k) by just w^k , since we are only consider unicast flows.

To analyze the multi-hop setting, we first start by analyzing a line network with a single source and a single destination. Consider the line network $G = (V, E)$, where $V = \{1, \dots, N\}$ and $E = \{(1, 2), (2, 3), \dots, (N-1, N)\}$ denote the N nodes and $N-1$ links, respectively. The network contains a single flow with source node $s = 1$ and this flow has a destination node $D_1 = \{N\}$. The commissioned nodes include all other nodes, to forward updates from the source to the destination, i.e. $C_1 = \{2, 3, \dots, N-1\}$. The source s generates fresh update packets that are transmitted over the line network to reach the destination node N .

In Section III-A, we saw that every stationary policy π is associated with link activation frequencies f_{ij}^k . For this simple line network, we now characterize the average age at the destination node N as a function of the link activation frequencies.

Theorem 1: If $f_e > 0$ is the link activation frequency for link $e \in E$ under a stationary policy π then the average age at node N is given by

$$A_{\text{ave}} = \lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T A_N(t) \right] = \sum_{e \in E} \frac{1}{\gamma_e f_e}, \quad (6)$$

where γ_e denotes the channel state probability for link $e \in E$.

Proof: See Appendix B. ■

Let π be a stationary randomized policy with link activation frequencies $\{\mathbf{f}^k\}_k$. Applying Theorem 1, we can write the average age for flow k to be:

$$A_{\text{ave}}^k = \lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T A_{d_k}^k(t) \right] = \sum_{e \in p^k} \frac{1}{\gamma_e f_e^k}. \quad (7)$$

This is because the set of nodes $\{k\} \cup C_k \cup \{d_k\}$ form a path from the source k to the destination node d_k . The weighted average age (in (2)) can be written as:

$$A_{\text{ave}} = \lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T \sum_{k=1}^K w^k A_{d_k}^k(t) \right], \quad (8)$$

$$= \sum_{k \in [K]} \sum_{e \in p^k} \frac{w^k}{\gamma_e f_e^k}. \quad (9)$$

The average age minimization problem can be written as:

$$\begin{aligned} & \text{Minimize}_{\mathbf{x} \in \mathbb{X}} \sum_{k \in [K]} \sum_{e \in p^k} \frac{w^k}{\gamma_e f_e^k}, \\ & \text{subject to } \mathbf{f} = M\mathbf{x}. \end{aligned} \quad (10)$$

We now observe that (10) can be viewed as an equivalent single-hop age problem.

Lemma 1: The multihop average age minimization problem over stationary randomized policies given by (10)) can be viewed as an equivalent single-hop age problem.

Proof: See Appendix C. ■

In this section, we developed a procedure to find age optimal stationary randomized policies in settings with known single commissioned paths between sources and destinations. These policies are easy to implement and analyze. They further allow us to derive closed form expressions of average age and provide performance guarantees. However, as we will see in the coming sections, these policies can only be used in limited settings and their performance is significantly far from optimal in practice.

IV. AGE DIFFERENCE POLICY

We now discuss a heuristic policy that yields a much lower average age, in practice, than even the optimal stationary randomized policy. We are still restricted to the setting with only unicast flows and known commissioned paths for each source-destination pair.

The basic idea is as follows: *in the propagation of information updates, it is important to keep the age differential between two neighboring nodes to as small a value as possible.*

Example: To illustrate this, consider a line network with N nodes and $N - 1$ links, with the single source and destination node placed at the two ends of the network. Assume all other nodes are commissioned for sending updates from the source to the destination node. Let the probability of successful transmission γ_e be 1 and the interference constraint be such that a transmission can occur only on one of the $N - 1$ links, at any given time.

For this network, we can deduce from Section III that the average age, given by $A_{\text{ave}} = \sum_{j=2}^N \frac{1}{f_{e_j}}$, is minimized at $f_{e_j} = 1/(N - 1)$ for all j and equals $A_{\text{ave}}^* = (N - 1)^2 = \mathcal{O}(N^2)$.

Now, consider a scheduling policy that works to minimize the age differential between any two nodes that share a link. This is done by scheduling the link e_j that has the maximum age differential. We schedule link e_{j^*} , at time t , such that

$$j^* = \arg \max_j [A_j(t) - A_{j-1}(t)]^+. \quad (11)$$

It can be deduced that under this scheduling policy over the line network the age at the destination node records the periodic pattern: $\{N - 1, N, N + 2, \dots, 2(N - 1), N - 1, N, N + 2, \dots, 2(N - 1), N - 1, N, \dots\}$. Thus, the average age at the destination node can be computed to be $A_{\text{ave}} = (N - 1) + N/2 = \mathcal{O}(N)$. Note that this is a significant improvement over the average age optimal stationary randomized policy.

Age-Difference Policy. We now articulate this age-difference heuristic for the general multi-flow, multi-hop network. Let $\Delta_{ij}^k(t)$ be the age difference weight for link (i, j) and flow k given by

$$\Delta_{ij}^k(t) \triangleq w_{ij}^k S_{ij}(t) [A_j^k(t) - A_i^k(t)]^+, \quad (12)$$

for all $j \in C_k \cup D_k$, $i \in \{k\} \cup C_k$, and flow $k \in [K]$. Note that w_{ij}^k denote the flow k weights for all the commissioned and the destination nodes. The w_{ij}^k s, for the destination nodes

$j \in D_k$, were defined in the average age cost function in (2). For all commissioned nodes, these weights are set to 1, i.e.

$$w_j^k = \begin{cases} w_j^k & \text{if } k \in D_k \\ 1 & \text{otherwise.} \end{cases}$$

Note that this choice is heuristic, and other sets of weights might lead to slightly different scheduling behavior.

The *age-difference* policy schedules a feasible set $m(t)$ that maximizes the age-difference weight, namely:

$$m(t) = \arg \max_{m \in \mathcal{A}} \sum_{(e,k) \in m} \Delta_e^k(t). \quad (13)$$

Result. We now show that the age-difference policy is in fact a one-step greedy policy for the average age minimization problem in (2), albeit with a modified age cost per time-slot.

Lemma 2: The age-difference policy is a myopic greedy policy in minimizing the average age

$$A_{\text{ave}} = \lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T \sum_{k=1}^K \sum_{j \in C_k \cup D_k} w_j^k A_j^k(t) \right]. \quad (14)$$

Proof: See Appendix D. ■

The average age in (14), differs from the average age defined in (2), in that it takes into account the age of also the commissioned nodes, for every flow k .

Greedy/myopic policies with similar structures have been shown to be constant factor optimal in the single-hop AoI minimization setting in [15] and [16].

The age-difference policy overcomes a key limitation of the stationary randomized policy, i.e. better performance in practice. However, it is still restricted to settings with a) only unicast flows, and b) source-destination paths that are fixed and known beforehand, and c) weighted sum AoI cost.

In the following section, we develop a general policy design framework that can address settings without any of these limitations.

V. AGE DEBT POLICY

In this section, we develop the age-debt framework for AoI minimization, based on ideas from Lyapunov optimization. To do so, we first introduce the notions of age-achievability and age debt virtual queues. We then show how stabilizing this network of virtual queues leads to minimization of AoI. Finally, we use quadratic Lyapunov drift to propose a heuristic scheme to achieve this stabilization in general multi-hop networks.

Note that in this section, we consider settings with a) general increasing cost functions of AoI, b) no knowledge of fixed routing paths, i.e. the scheduler also needs to make routing decisions and c) unicast, multicast and broadcast flows in the same network. The general AoI optimization problem can be formulated as:

$$\pi^* = \arg \min_{\pi} \left(\lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T \sum_{k=1}^K \sum_{j \in D_k} B_j^k(t) \right] \right), \quad (15)$$

where $B_j^k(t)$ are the effective age processes and $\pi(t) \in \mathcal{A}, \forall t$. This setting is more general than the ones considered in Section III and Section IV.

A. Age Debt

We start by assuming that we have been given a target value of time average age cost for each source-destination pair; denoted by α_j^k for the source-destination pair (k, j) . We aggregate the target values associated with each source-destination pair in the vector α . For any such target vector α , we define the notion of age-achievability below.

Definition A vector α is **age-achievable** if there exists a feasible network control policy π such that

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T B_j^k(t) \leq \alpha_j^k, \quad \forall j \in D_k, \forall k \text{ w.p. 1.} \quad (16)$$

In other words, a vector α is age-achievable if the time-average of the effective age process for *every* source-destination pair (k, j) is upper bounded by the target value α_{kj} , under some feasible network control policy.

Note that the combination of general cost functions and achievability targets allows us to capture very general freshness requirements which might be useful in practical system specifications. For example, if an application requires that the empirical distribution of the age process $A_j^k(t)$ should satisfy $\mathbb{P}(A_j^k(t) \geq M) \leq \epsilon$, then we can capture this by setting the cost function $g_j^k(h) = \mathbf{1}_{\{h \geq M\}}$ and the corresponding target to be $\alpha_j^k = \epsilon$.

We now define a set of virtual queues called age-debt queues for every source-destination pair (k, j) . These queues measure how much the effective age process exceeds its target value α_{kj} , summed over time. Our definition of debt is inspired by the notion of throughput debt as introduced in [33].

Definition Given a target vector α , the **age debt queue** for source-destination pair (k, j) at time t , given by $Q_j^k(t)$, evolves as

$$Q_j^k(t+1) = \left[Q_j^k(t) + B_j^k(t+1) - \alpha_j^k \right]^+, \quad \forall j \in D_k, \text{ and } \forall k \in \{1, \dots, K\}. \quad (17)$$

To complete the definition, each age debt queue starts at zero, i.e. $Q_j^k(0) = 0, \forall j, k$.

We now introduce a notion of stability for these age debt queues. This is similar to how rate stability is typically defined in queueing networks [34].

Definition We say that the network of age debt queues is **stable** under a policy π and a given target vector α if the following condition holds:

$$\lim_{T \rightarrow \infty} \mathbb{E} \left[\sum_{k=1}^K \sum_{j \in D_k} \frac{Q_j^k(T)}{T} \right] = 0, \quad (18)$$

where the expectation is taken over the randomness in the channel processes and the scheduling policy π .

We also establish an equivalence relationship between age-achievability of a vector α and the stability of the corresponding network of age debt queues.

Lemma 3: A target vector α is age-achievable if and only if there exists a network control policy π , that stabilizes the network of source-destination age debt queues.

Proof: See Appendix E. ■

Next, we define a debt-stable scheduling policy. Such a policy takes a target vector α as an input and stabilizes the network of corresponding age debt queues.

Definition A debt-stable scheduling policy π stabilizes the set of age-debt queues for any given target vector α that is age-achievable.

The notions introduced until now effectively allow us to convert the minimum age cost problem described by (15) into a network stability problem. Suppose π^* is a solution to the optimization problem (15). Further, suppose that the time average of the effective age process for pair (k, j) under π^* is given by

$$\lim_{T \rightarrow \infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T B_j^k(t) \right] = \alpha_j^k, \forall (k, j). \quad (19)$$

Clearly, if we have oracle access to an optimal age cost vector $\alpha^* = \{\alpha_j^k\}_{(k,j)}$ and know how to design a debt-stable policy then we can perform minimum age cost scheduling. If the debt-stable policy is much lower in computational complexity than solving (15) directly, then we can also solve (15) at the same lower complexity (assuming oracle access to α^*). We now discuss a heuristic approach to designing debt-stable policies.

B. Lyapunov Drift Approach

1) Single-Hop Broadcast: We first consider the special case of single-hop broadcast networks. This setting is easier to analyze since it only requires scheduling and no routing and it also highlights key structural properties of our proposed policy.

Consider a N node star network where each of the nodes $1, \dots, N-1$ has an edge to node N . These nodes wish to send packets to the central node N . The edges are numbered $e_1, \dots, e_{N-1}$. Due to broadcast interference constraints, only one node can transmit in any given time-slot. Since the destination for every flow is N , we can drop the destination in our notation. The age evolution is given by

$$A_i(t+1) = \begin{cases} A_i(t) + 1, & \text{if } U_{e_i}(t) S_{e_i}(t) = 0 \\ 1, & \text{if } U_{e_i}(t) S_{e_i}(t) = 1. \end{cases} \quad (20)$$

Given an age-cost function $g_i(A_i(t))$ and a corresponding target value α_i , the debt queue evolution for node i is given by:

$$Q_i(t+1) = \left[Q_i(t) + g_i(A_i(t+1)) - \alpha_i \right]^+. \quad (21)$$

Given a target vector α , we will use a Lyapunov drift based scheduling scheme to try and achieve debt stability. To do so, we first define a Lyapunov function for our system of virtual queues:

$$L(t) \triangleq \sum_{i=1}^{N-1} Q_i^2(t). \quad (22)$$



Fig. 2. Example of a line network with a unicast flow from node 1 to node 3.

Using this Lyapunov function, we then define the *age debt scheduling policy* π^{AD} as:

$$\pi^{\text{AD}}(t) = \underset{a \in \mathcal{A}}{\operatorname{argmin}} \left(\mathbb{E}[L(t+1) - L(t)] \right), \quad (23)$$

where the expectation is taken over the randomness in channel reliabilities $\mathcal{S}(t)$.

In the following remark, we consider a variant of the age-debt policy that minimizes an upper-bound on the Lyapunov drift instead of the actual Lyapunov drift as in (23). This is similar to the upper-bound drift minimization used in policies such as max-weight [35] and allows us to compare the structure of age-debt to previously proposed policies in literature.

Remark 1: Suppose that the links between each source i and the destination N are i.i.d. Bernoulli w.p. γ_i in every time-slot. Further, if each age cost function $g_i(\cdot)$ is upper bounded by a large constant D , then the policy $\pi(t)$ below minimizes an upper bound on the Lyapunov drift in every time-slot.

$$\pi(t) = \underset{i \in 1, \dots, N-1}{\operatorname{argmax}} \left(\gamma_i Q_i(t) (g_i(A_i(t) + 1) - g_i(1)) \right). \quad (24)$$

Proof: See Appendix F. ■

In other words, an approximate drift minimizing policy chooses the source with the largest product of link reliability, current age debt and current age cost. This structure of the age-debt policy can be contrasted with the max-weight policy proposed in [15] which chooses the source with the largest value of $\gamma_i w_i A_i(t) (A_i(t) + 2)$ given weights w_i . Similarly, the Whittle index policy proposed in [24], chooses the source with the largest value of $W_i(A_i(t))$, where $W_i(\cdot)$ is Whittle-index corresponding to the age cost $f_i(\cdot)$.

Note that to compute $\pi^{\text{AD}}(t)$, the scheduler needs to iterate over the set of sources only once. So the per slot computational complexity of this policy grows linearly in N . This is similar to the complexity of the Whittle index policy proposed in [15] and [24] and the max-weight policies proposed in [15] and [16]. By contrast, a dynamic programming approach to solve (15) directly has per slot computational complexity that grows exponentially in N . This highlights the key strength of our approach. If the scheduler has some way to set the targets for each source optimally, then the age debt policy is a good low complexity heuristic for age minimization.

2) *General Networks:* The general multihop setting is more challenging. Simply using one-slot Lyapunov drift to try and achieve debt stability does not work directly in the multihop setting. We highlight this with a simple example.

Consider the three node network described in Figure 2 with a single unicast flow from node 1 to node 3. The interference constraint enforces that only one of the two edges a and b can be activated in any time-slot. Suppose that we are interested in

minimizing the time average of the age process $A_3^1(t)$. Given a target value α_3^1 , we set up the age debt queue as follows:

$$Q_3^1(t+1) = \left[Q_3^1(t) + A_3^1(t+1) - \alpha_3^1 \right]^+. \quad (25)$$

We will try to use the one slot Lyapunov drift minimizing policy to stabilize $Q_3^1(t)$ in this network. To do so, we solve the following optimization in every time-slot:

$$\pi^{\text{AD}}(t) = \underset{x \in \{a, b\}}{\operatorname{argmin}} \left(\mathbb{E}[(Q_3^1(t+1))^2 - (Q_3^1(t))^2] \right). \quad (26)$$

At $t = 1$, activating either edge a or edge b has no effect on the debt $Q_3^1(2)$ since node 2 does not have any packet from node 1. If we break ties in favour of edge b , then it is activated but no new packet is delivered to node 3. At $t = 2$, since node 2 still does not have any new update from node 1, no action taken can affect the debt $Q_3^1(3)$. Using the same tie-break rule, we would again schedule edge b . This process keeps on repeating and the age debt queue Q_3^1 blows up irrespective of the value of α_3^1 , even though the age optimal policy in this setting is to simply alternate between a and b in every time-slot.

The example above illustrates why one-slot Lyapunov drift based techniques fail in stabilizing debt queues in multi-hop networks. The policy designer using Lyapunov drift is constrained to optimizing *only one time-step into the future*, similar to a greedy policy. So, if every possible scheduling and routing action has no effect on the age debt queues in the immediate next time-slot, the one step drift minimizing procedure does not provide any information on which action should be chosen to stabilize the debt queues.

This suggests that to be able to use one-slot drift minimizing techniques for stability, there should be a virtual queue for every intermediate node that tracks both the current age debt at the destination and the potential reduction in debt at the destination upon forwarding a fresh packet. If we can set up such queues, then large values of debt at intermediate nodes would lead to fresh packets being sent to the next hops via one-slot drift minimizing actions, eventually reaching the destination and stabilizing the age debt queues.

Let $Q_j^{k \rightarrow i}(t)$ denote such a debt queue corresponding to flow (k, j) at an intermediate node i . These additional queues at every intermediate node combined with the original debt queues form our virtual network. The Lyapunov function that we use for scheduling and routing is given by:

$$L(t) \triangleq \sum_{k=1}^K \sum_{j \in D_k} \left((Q_j^k(t))^2 + \sum_{i \notin D_k, i \neq k} (Q_j^{k \rightarrow i}(t))^2 \right) \quad (27)$$

‘The Age Debt scheduling and routing policy is to choose the activation set and corresponding flows that minimizes the expected Lyapunov drift.

$$\pi^{\text{AD}}(t) = \underset{a \in \mathcal{A}}{\operatorname{argmin}} \left(\mathbb{E}[L(t+1) - L(t)] \right), \quad (28)$$

where the expectation is taken over the randomness in channel reliabilities $\mathcal{S}(t)$.

3) *Intermediate Debt Queues*: We now discuss how to set up the age debt queues $Q_j^{k \rightarrow i}(t)$ for intermediate nodes to augment the original network of queues. Note that there are no intermediate nodes for broadcast flows since every node other than the source is a destination.

Consider a source-destination pair (k, j) for a unicast/multicast flow k and an intermediate node i that is not a destination for the flow originating at k . We want to set up the age debt queue $Q_j^{k \rightarrow i}(t)$ at i for the pair (k, j) . We maintain an age process for flow k at node i , even though there is no associated cost or target value for this age process.

$$A_i^k(t+1) = \begin{cases} \min(A_i^k(t), t - t_g) + 1, & \text{if update generated} \\ & \text{at time } t_g \text{ is delivered at time } t. \\ A_i^k(t) + 1, & \text{if no new delivery at time } t. \end{cases} \quad (29)$$

Here $A_i^k(t)$ measures how old the information at node i is regarding node k . We split the debt queue's evolution into two cases.

Case 1: When node i forwards a flow k packet on a set of adjacent links L . Let h_{ij}^L be the minimum number of hops it takes to reach node j from node i , where the first hop can only include edges in the set L . Here, h_{ij}^L measures the minimum delay with which the packet that was forwarded by i gets delivered at j . The age debt queue $Q_j^{k \rightarrow i}(t)$, when node i is forwarding a flow k packet along the link set L , evolves as:

$$Q_j^{k \rightarrow i}(t+1) = \left[Q_j^{k \rightarrow i}(t) + g_j^k \left(\min\{A_i^k(t), A_j^k(t)\} + h_{ij}^L \right) - \alpha_j^k \right]^+ \quad (30)$$

This measures the most optimistic change in age debt possible at the destination using the current packet transmission from node i .

Case 2: When node i does not forward a packet from node k along any of its adjacent edges, then the age debt queue evolves as below.

$$Q_j^{k \rightarrow i}(t+1) = \left[Q_j^{k \rightarrow i}(t) + B_j^k(t+1) - \alpha_j^k \right]^+ \quad (31)$$

This means that the intermediate queue simply tracks the change in debt at the destination when it is not forwarding a relevant packet. If the destination is not receiving fresh packets from anywhere in the network then this would increase the intermediate debt queue.

Thus, the debt at an intermediate node i for a source-destination pair (k, j) blows up if (a) either the destination has not received fresh packets for a long time and node i did not forward any packets from k (i.e. (31)) or if (b) node i keeps forwarding stale packets from k (i.e. (30)). A drift minimizing policy will then try to ensure that either the destination debt queue is small, or node i forwards fresh packets of flow k towards the destination.

C. An Example

Consider the five node network depicted in Fig. 3. We consider two flows in this network - a multicast flow from

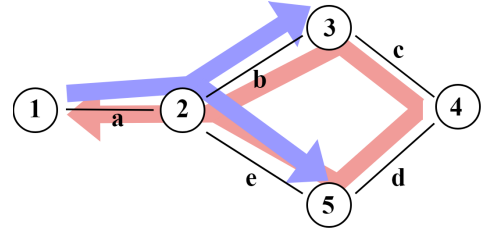


Fig. 3. Example of a five node multihop network with two competing flows.

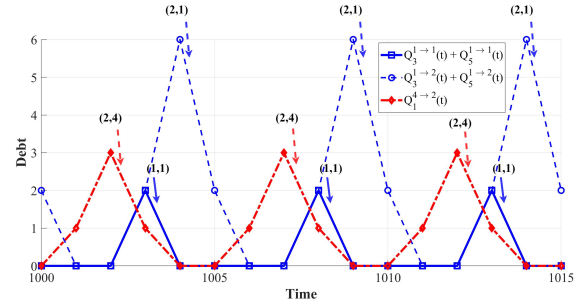


Fig. 4. Evolution of three debt queue quantities with time, along with scheduling decisions involving link a .

source 1 to destination nodes 3 and 5 and a unicast flow from source 4 to destination node 1. We further consider interference constraints such that every node interferes with every other node, so in any given time-slot only one node can transmit successfully. Further, the transmitting node can only transmit a single flow's packet on all of its links, i.e. it cannot choose to send different flow packets on different links. Thus, the available choice of actions in each time-slot is to decide which node gets to transmit and regarding which flow.

Given average AoI targets α_3^1, α_5^1 and α_1^4 , we will use our proposed age debt scheme to achieve them. Observe that edge a between nodes 1 and 2 must be used by both flows and hence acts like a bottleneck link. One possible schedule that we might want to replicate is to send packets for the first flow, and then send packets for the second flow, and keep alternating. If we represent actions as a tuple (i, j) denoting that node i forwards an update regarding node j , then this scheduling policy looks like $(1, 1) \rightarrow (2, 1) \rightarrow (4, 4) \rightarrow (5, 4) \rightarrow (2, 4)$. It is easy to see that the average AoIs achieved by this policy are $\alpha_3^1 = 4.0, \alpha_5^1 = 4.0$ and $\alpha_1^4 = 5.0$.

Providing these targets to the age debt queues, we confirm via simulation that our one slot Lyapunov drift minimization method does indeed achieve the required average AoIs. To understand how age debt makes scheduling decisions, we will focus on the bottleneck link a . Recall that $Q_j^{k \rightarrow i}$ is the intermediate debt queue at a forwarding node i for the source k and destination j . The debt queues that primarily influence which flow gets transmitted on link a are $Q_3^{1 \rightarrow 1}(t), Q_5^{1 \rightarrow 1}(t), Q_3^{1 \rightarrow 2}(t), Q_5^{1 \rightarrow 2}(t)$, and $Q_1^{4 \rightarrow 2}(t)$.

Given $\alpha_3^1 = 4.0, \alpha_5^1 = 4.0$ and $\alpha_1^4 = 5.0$, we will plot three debt queue quantities once the age debt policy has reached steady state. Specifically, we plot $Q_3^{1 \rightarrow 1}(t) + Q_5^{1 \rightarrow 1}(t), Q_3^{1 \rightarrow 2}(t) + Q_5^{1 \rightarrow 2}(t)$, and $Q_1^{4 \rightarrow 2}(t)$.

There are three competing actions that the scheduling policy can take for the link a . These are (1, 1) - node 1 broadcasts an update about itself along link a ; (2, 1) - node 2 broadcasts an update about node 1 along links a, b and e ; and (2, 4) - node 2 broadcasts an update about node 4 along links a, b and e . In Fig. 4, we plot the three debt queue quantities of interest along with timeslots in which scheduling decisions involving link a were made.

We observe that the relative ordering of the three debt quantities explains why each scheduling decision is made. When $Q_3^{1 \rightarrow 1}(t) + Q_5^{1 \rightarrow 1}(t)$ is the largest, it suggests that sending an update from node 1 to node 2 along link a is the most valuable action. Similarly, when $Q_3^{1 \rightarrow 2}(t) + Q_5^{1 \rightarrow 2}(t)$ is the largest, it suggests that sending an update regarding node 1 from node 2 to nodes 3 and 5 is the most valuable action (and due to the interference constraint, it blocks any other node from using the link a). When $Q_1^{4 \rightarrow 2}(t)$ is the largest, it suggests that sending an update regarding node 4 from node 2 to node 1 along link a is the most valuable action.

Note that scheduling decisions are not explicitly made based on relative ordering of the debt queues, but by finding the action that has the largest negative Lyapunov drift. We use the relative ordering to provide an intuitive explanation of age debt. Importantly, the evolution of the debt queues and the Lyapunov drift approach together lead to a scheduling policy that achieves the desired AoI targets. Adjusting the targets and/or AoI cost functions, the system designer can tradeoff the freshness of one flow against another.

D. Choosing Target Vectors

In the preceding sections, we have developed a general framework of age achievability where given a target average age cost for every source-destination pair, we formulate a corresponding network stability problem and attempt to solve it via one slot Lyapunov drift minimization. In this section, we discuss how to choose the right target vectors, such that they lead to minimum sum age cost.

In the absence of an optimization oracle that provides access to α^* or a system administrator who specifies average age cost targets based on the underlying application requirements, we develop a simple heuristic to dynamically update α in order to optimize utility based on the state of the underlying debt queues.

The following optimization problem needs to be solved to find the best target vector α^* .

$$\begin{aligned} \argmin_{\alpha} \quad & \left(\sum_{k=1}^K \sum_{j \in D_k} \alpha_j^k \right), \\ \text{s.t. } \quad & \alpha \text{ is age-achievable.} \end{aligned} \quad (32)$$

Note that this problem has the same optimal value as (15).

1) *Gradient Descent*: We want to use a gradient descent like approach to solve (32) and find α^* . The problem with doing so is that we do not have a simple characterization of the age-achievability region or a low complexity method to test whether a vector is achievable or not.

To resolve this, we use Lemma 3. If the network of source-destination age debt queues is unstable for a given

value of α , then α lies outside the age-achievability region. This immediately suggests the gradient descent like approach described in Algorithm 1.

Algorithm 1 Age Debt - Gradient Descent

Input : epoch size W , number of epochs E , step-size $\eta > 0$, threshold $\epsilon > 0$, initialization $\alpha(1)$

```

1 while  $e \in 1, \dots, E$  do
2   Set up age debt queues using  $\alpha(e)$  and initialize each queue to 0
3   while  $t \in 1, \dots, W$  do
4     Schedule and route using age debt
4      $\pi^{\text{AD}}(t) = \operatorname{argmin}_{a \in \mathcal{A}} \left( \mathbb{E}[L(t+1) - L(t)] \right)$ ,
5   end
6   if  $\exists \text{ flow } k \text{ and } j \in D_k \text{ s.t. } Q_j^k(W) > \epsilon W$  then
7     Increase target values for unstable queues:
8      $\alpha_j^k(e+1) = \alpha_j^k(e) + \eta, \forall (k, j) \text{ s.t. } Q_j^k(W) > \epsilon W$ 
9     Other targets remain unchanged:
10     $\alpha_j^k(e+1) = \alpha_j^k(e), \forall (k, j) \text{ s.t. } Q_{kj}(W) \leq \epsilon W$ 
11  end
12  else
13    Update all target values using gradients:
13     $\alpha_j^k(e+1) = \alpha_j^k(e) - \eta, \forall (k, j)$ .
14  end
15 end
```

The algorithm above runs the age debt policy for epochs of length W time-slots. Within an epoch, the target vector α remains fixed. At the end of the epoch, we use the value of the source-destination age debt queues $Q_j^k(\cdot)$ to update the corresponding targets. If the network has at least one queue with debt larger than a threshold, it suggests that the current vector is not achievable. So, we increase the values of α for the source-destination pairs with large values of debt. If the network has all queues with debt below a threshold, the current vector is likely achievable. So, we update the entire target vector using gradient descent. Note that this approach takes a large number of time-slots to converge to a good candidate target vector α .

2) *Flow Control*: Another way to dynamically set the target vectors is to take a flow control approach for solving the optimization problem (32), similar to [35]. Algorithm 2 describes the details.

The flow control based age debt policy tries to tradeoff between the stability of the queueing network and the optimization of targets using a parameter $V > 0$. In every time-slot, the flow control optimization sets the target α for the next time-slot and then the scheduling and routing decisions are computed by minimizing Lyapunov drift.

The update optimization in step 4 of Algorithm 2 can be simplified to the rule below:

$$\alpha_j^k(t+1) = \begin{cases} \alpha_{\max}, & \text{if } Q_j^k(t) > V \\ 1, & \text{if } Q_j^k(t) \leq V, \end{cases} \forall (k, j). \quad (33)$$

Algorithm 2 Age Debt - Flow Control

Input : parameter $V > 0$, upper bound $\alpha_{\max}$,
initialization $\alpha(1)$

```

1 while  $t \in 1, \dots, T$  do
2   Use  $\alpha(t)$  to update debt queue values at time  $t$ 
3   Update  $\alpha$  by solving the optimization below:
4

$$\alpha(t+1) = \underset{\alpha}{\operatorname{argmin}} \left( \sum_{k=1}^K \sum_{j \in D_k} V \alpha_j^k - \alpha_j^k Q_j^k(t) \right),$$


$$\text{s.t. } \alpha \geq 1, \alpha \leq \alpha_{\max}.$$

5   Use  $\alpha(t+1)$  to compute the scheduling and routing
    decision that minimizes drift:

$$\pi(t) = \underset{a \in \mathcal{A}}{\operatorname{argmin}} \left( \mathbb{E}[L(t+1) - L(t)] \right)$$

6 end
```

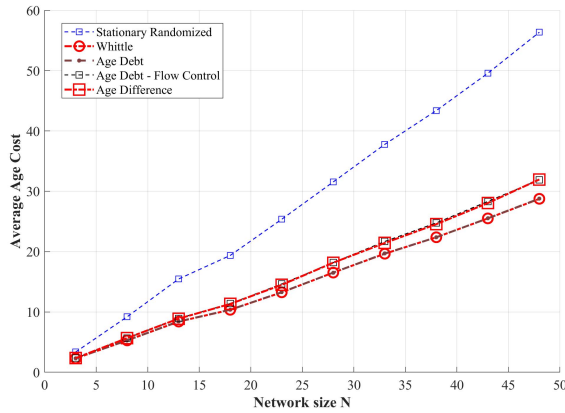


Fig. 5. Weighted-sum AoI minimization in broadcast networks with unreliable channels.

Thus, instead of converging to a target vector as in the case with gradient descent, the flow control approach dynamically switches the value of targets in every time-slot. This means we do not need to wait a long period of time for convergence. When current debts are high, future targets are set to be high pushing the debts lower. Similarly, when the current debts are low, future targets are also set low, pushing the debts higher. The parameter V decides the threshold between high and low values of the debt queues.

VI. NUMERICAL RESULTS

First, we consider the weighted-sum AoI problem in single-hop broadcast networks with unreliable channels. There are N nodes in the network and the weight of the i th node w_i is set to i/N . Link connection probabilities are chosen uniformly from the set $[0.6, 1]$. Figure 5 plots the performance of the age debt policy, the age difference policy, and the optimal stationary randomized policy along with the max-weight and Whittle index policies proposed in [15] which are known to be close to optimal.

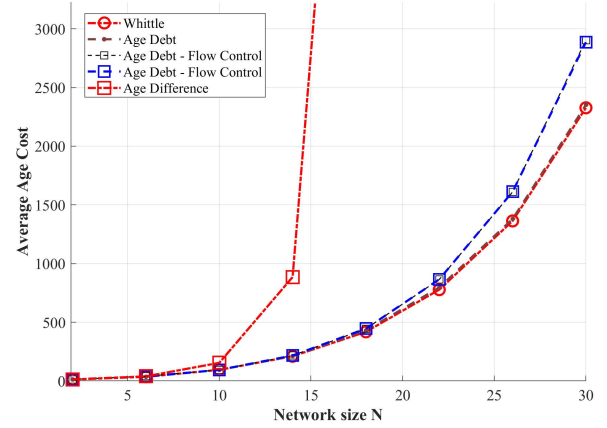


Fig. 6. Functions of Age minimization in broadcast networks with reliable channels.

First, we observe that the optimal stationary randomized policy performs much worse than the other classes of policies. Age difference performs better than the randomized policy but not as well as the Whittle-index or max-weight policies. We further observe that when the age debt policy is provided the max-weight average cost as the target vector, it replicates near optimal performance. Also, the flow control and gradient descent versions of age debt have a small gap to the max-weight/Whittle policies despite not having access to α beforehand and perform as well as the age-difference policy.

Next, we consider general functions of age minimization in the single-hop wireless broadcast setting. There are N nodes in the network and the cost of AoI for each node is chosen from the set of functions $\{15A(t), e^{A(t)}, (A(t))^2$ and $(A(t))^3\}$. Figure 6 plots the performance of the age-debt policy and its variants along with the age difference policy and the Whittle index policy proposed in [24]. As for the linear AoI case, we observe that age debt is able to replicate the Whittle policy's performance when provided its average cost as the target vector. The flow control and gradient descent variants are also only a small gap away in performance without knowing α beforehand. On the other hand, the age difference policy performs much worse and the AoI cost rapidly grows very large even for moderate N . This is because the age difference policy is not designed to handle general AoI cost functions, so even though it tries to keep the AoIs small for all nodes, their actual impact to cost can become very large. It was also shown in [24] that even the optimal stationary randomized policy can have unbounded AoI cost for systems as small as $N = 2$, given nonlinear AoI cost functions. So, we do not plot its performance in this scenario.

We also look at the functions of age problem with $N = 4$ in more detail. The age cost functions for each node are as follows $f_1(A_1(t)) = 15A_1(t)$, $f_2(A_2(t)) = e^{A_2(t)}$, $f_3(A_3(t)) = (A_3(t))^2$ and $f_4(A_4(t)) = (A_4(t))^3$. First, we use dynamic programming to compute the optimal policy π^* which minimizes average age cost. The time average age costs under this policy are given by $\alpha_1^* = 45.0$, $\alpha_2^* = 14.52$, $\alpha_3^* = 17.20$, and $\alpha_4^* = 11.0$, while the total sum cost is 87.72. Using these

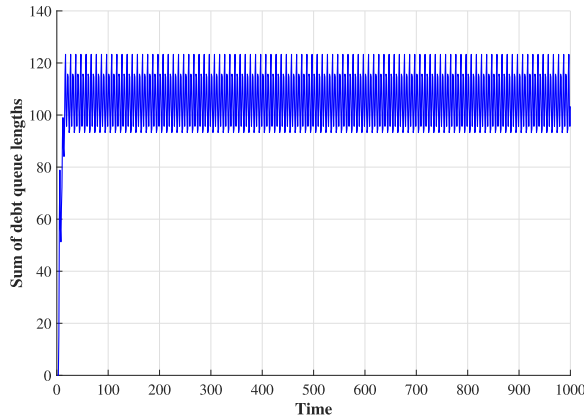


Fig. 7. Sum of virtual debt queues vs time.

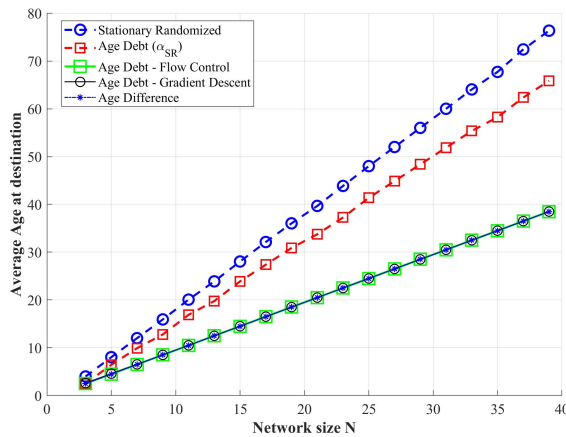


Fig. 8. A single unicast flow on a line network (neighboring nodes interfere).

as target values, we set up debt queues and implement the age-debt policy.

Figure 7 plots the sum of the 4 age debt queues $\sum_{i=1}^4 Q_i(t)$ under the age-debt policy implemented using the optimal α^* from above. We observe that the age debt policy indeed stabilizes the debt queues since queue lengths don't grow with time. As a corollary, it also achieves age cost optimality in this setting. On the other hand, the Whittle index policy from [24] achieves a total sum cost of 88.34, a fixed but small distance away from the optimal cost of 87.72. This suggests that age-debt might be a way to achieve exact optimality instead of near optimality when access to α^* is available.

Next, we consider scheduling for a single unicast flow on the line network. Consider N nodes arranged in a line network from 1 to N . Node 1 wants to send packets to node N , however not all nodes can transmit simultaneously. We consider a simple interference constraint - in any given time-slot either all even numbered nodes or all odd numbered nodes can forward packets. This ensures that no two adjacent nodes send interfering transmissions. Figure 8 plots the performance of age-debt and its flow-control and gradient-descent variations along with the optimal stationary randomized policy proposed in Section III and the age difference policy proposed in Section IV. We observe that age-debt outperforms the

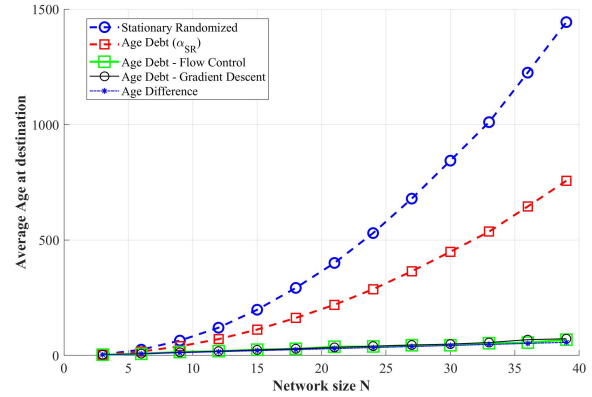


Fig. 9. A single unicast flow on a line network (all nodes interfere).

stationary randomized policy despite using its average cost α_{SR} as the target vector. The dynamic variants of age-debt significantly outperform the stationary randomized policy and match the performance of the age-difference policy. We also note that the gap in performance would increase in settings with multiple flows and paths available which age-debt can utilize for routing, unlike the stationary randomized and age difference approaches.

We also consider a different kind of interference constraint in the same line network example. Now, all nodes interfere with one another, and only one node can transmit successfully in any given time-slot. We plot the performance of the optimal stationary randomized policy, the age-debt policy (provided α_{SR}), our age-debt variants without any knowledge of α and the age difference policy against the number of nodes in the system in Figure 9. We again observe a large gap in performance between the optimal randomized policy and our proposed methods. This is consistent with the line network AoI analysis from Section IV, where we showed that the best stationary randomized policy has performance that is $O(N^2)$ while the age difference policy has performance $O(N)$. We also observe that the age-debt variants match the performance of the age difference policy, which can be shown to be exactly optimal in this single source line network setting.

Finally, we consider average age minimization for all-to-all broadcast flows in multihop networks similar to [28]. Note that this is a broadcast setting that requires both scheduling and routing decisions to be made, so we cannot use the stationary randomized or age difference policies developed in Sections III and IV. We consider all possible connected network topologies with 5 or 6 nodes (a total of 133 graphs). Figure 10 plots the performance of the age-debt policy and its variants along with the near optimal minimum connected dominating set (MCDS) based scheme proposed in [28] for each of these networks. The x-axis represents the graph labels numbered from 1 to 133, sorted according to the average age achieved by the MCDS scheme.

We observe that age-debt achieves the same performance as the MCDS scheme when provided its average cost as the target vector. Further, age-debt with flow control achieves performance that is very close to that of the MCDS scheme without

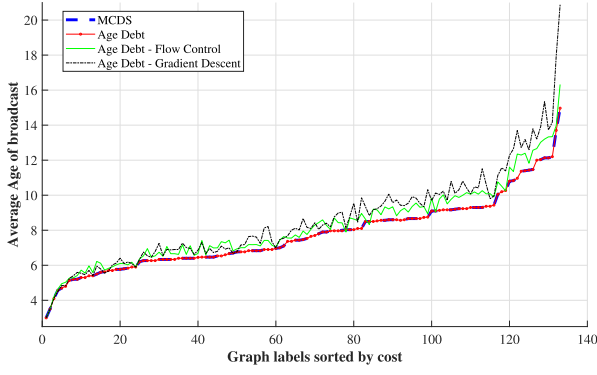


Fig. 10. Broadcast flows in multihop networks with 5 and 6 nodes.

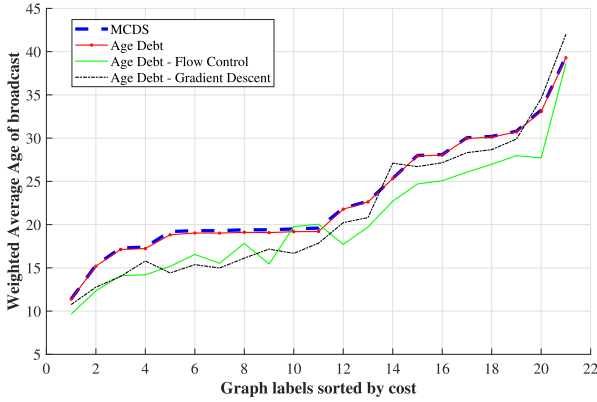


Fig. 11. Weighted Age minimization of broadcast flows in multihop networks with 5 nodes.

requiring knowledge of α . Importantly, the MCDS scheme can only be applied to this setting of all-to-all broadcast with one node transmitting at a time. Further, computing the optimal schedule using the MCDS scheme requires finding minimum size connected dominating sets, the complexity of which grows exponentially in the number of nodes.

We also consider the same broadcast setting but now with weighted-sum AoI as the minimization objective instead of just AoI. We consider all possible connected graphs with 5 nodes (21 in total). We set the importance weight of one node to 15 (giving it a higher priority) and the rest of the 4 nodes to 1. Figure 11 plots the performance of the MCDS scheme along with age-debt and its variants. As expected, age-debt policy replicates the performance of the MCDS scheme since it is provided the average age-cost realized by MCDS as the target. Interestingly, flow-control outperforms MCDS since it is able to adapt to a better target α in the presence of weights and asymmetry. This is consistent with the fact that the MCDS scheme is not designed for minimizing weighted-sum AoI. It also highlights the relative ease with which age-debt can be adapted to weights and general AoI cost functions.

Note that the complexity of implementing the flow-control scheme is polynomial in the network size per time-slot. This suggests that age-debt and its variants are a good candidate for low complexity near optimal age scheduling in general networks.

We also observe that the flow control variant of age-debt is the method of choice in the absence of known α . During our experiments, we found that the gradient descent variant has parameters that are hard to configure for networks of different sizes and takes a long time to converge. The flow-control method has just two parameters V and $\alpha_{\max}$ that are relatively easy to configure and do not require any time for convergence.

VII. CONCLUSION

We considered the problem of minimizing age-of-information freshness metrics for general multi-hop networks and proposed three classes of policies - stationary randomized, age-difference and age-debt. Through analysis and simulations, we compared the performance of these three policies with the best known policies previously known in literature for a wide variety of settings. Directions of future exploration involve 1) proving performance bounds for age-debt and its variants, and 2) considering distributed implementation, stochastic arrivals and time-varying network topologies.

APPENDIX

A. Stationary Randomized Policies in Single-Hop Networks

Consider the special case when we are interested in minimizing A_{ave} in a single hop network, with general interference constraints and multiple nodes sending updates to a base station (BS). Since each edge simply connects one node to the BS, it can only forward packets from that node. Thus, the notation for the link activation frequencies can be simplified from f_{ij}^k to f_e where e is the only edge connecting node e to the BS and also the only edge transmitting flow e packets. It is easy to see that for such a network, the weighted average age minimization problem can be written as:

$$\begin{aligned} & \text{Minimize} \sum_{e \in E} \frac{w_e}{\gamma_e f_e}, \\ & \text{subject to } \mathbf{f} = M\mathbf{x}. \end{aligned} \quad (34)$$

We, therefore, call it the single-hop age problem. As we discussed in Sec. III-B, all average age minimization problems - even though we are dealing with multi-hop flows - can be reformulated to look like (34).

B. Proof of Theorem 1

Consider the line network $G = (V, E)$, where $V = \{1, \dots, N\}$ and $E = \{(1, 2), (2, 3), \dots, (N-1, N)\}$ denote the N nodes and $N-1$ links, respectively. For convenience, we use e_j to denote the link $(j-1, j)$ and U_{e_j} and S_{e_j} to denote the transmission and channel state on $(j-1, j)$.

For this simple line network, we show that the age evolution in (1) can be simplified.

Lemma 4: The age evolution of (1) can be written as

$$A_j(t+1) = \begin{cases} A_{j-1}(t) + 1 & \text{if } U_{e_j}(t)S_{e_j}(t) = 1 \\ A_j(t) + 1 & \text{otherwise,} \end{cases}$$

for all $j \in C_1 \cup D_1 = \{2, 3, \dots, N\}$.

Proof: If $U_{e_j}(t)S_{e_j}(t) = 0$, then no successful transmission occurs over link e_j , and therefore, it follows from (1)

that $A_j(t+1) = A_j(t) + 1$. If $U_{e_j}(t)S_{e_j}(t) = 1$, then a successful transmission does occur over link e_j at time t , and two possibilities arise: either node $j-1$ has a fresh update, received since the last transmission over the link e_j , or it hasn't. If the node $j-1$ has a fresh update, then this fresh update packet is transmitted to node j at time t , and we get $A_j(t+1) = A_{j-1}(t) + 1$. If, on the other hand, node $j-1$ hasn't received a fresh update since the last transmission over link e_j , then we have $A_j(t) = A_{j-1}(t)$. This is because both nodes have the same update packet that was exchanged during the last transmission over link e_j . The age evolution, in the absence of a fresh update, therefore becomes $A_j(t+1) = A_j(t) + 1 = A_{j-1}(t) + 1$. ■

The age evolution equation in Lemma 4 is true irrespective of the scheduling policy.

Given any stationary randomized policy, we want to show that

$$\mathbb{E}[A_N(t)] = \sum_{e \in E} \frac{1}{\gamma_e f_e} = \sum_{j=2}^N \frac{1}{\gamma_{e_j} f_{e_j}}, \quad (35)$$

where f_{e_j} is the link activation frequency for link e_j and γ_{e_j} is the probability that the state of the link e_j is 1. The result (in (6)) follows from (35).

Let π be a stationary randomized policy, and f_{e_j} be the link activation frequency for link e_j under policy π , for all $j \in \{2, \dots, L\}$. Let $\alpha_j \triangleq \gamma_{e_j} f_{e_j}$. Further, let $\tau_j(t)$ denote the last instance, when a successful transmission occurred over link e_j . For example, if link activations occurred at time slots 2, 10, 14, and 21 then $\tau_j(t) = 10$ for all $t = 11, 12, 13$, and 14.

Since π is a stationary randomized policy, the (successful) inter-transmission times must be geometrically distributed with mean $\frac{1}{\alpha_j}$, for link e_j . The memoryless property, therefore, implies that

$$\mathbb{P}[\tau_j(t) = t - s] = \alpha_j (1 - \alpha_j)^{s-1}, \quad (36)$$

for all $s = 1, 2, \dots$. Thus, $\tau_j(t)$ has the same distribution as $t - X_{e_j}$, where X_{e_j} is a geometrically distributed random variable given by

$$\mathbb{P}[X_{e_j} = s] = \alpha_j (1 - \alpha_j)^{s-1}, \quad (37)$$

for all $s \in \{1, 2, \dots\}$, with mean

$$\mathbb{E}[X_{e_j}] = \frac{1}{\alpha_j}, \quad (38)$$

for all the links e_j in the network.

Node 1 being the active source, its age is always 0. Consider age at node 2. Since the source node always transmits fresh information, the age of node 2 is the just the time elapsed since the last successful transmission over the link e_2 . This is given by

$$A_2(t) = t - \tau_2(t), \quad (39)$$

as $\tau_2(t)$ was the last time a fresh update packet was sent to node 1.

Next, consider age at node 3. By Lemma 4, whenever a successful transmission occurs over link e_3 , node 3 resets its age to node 2's age $A_2(\cdot)$. Thus, $A_3(t)$ is given by

$$A_3(t) = t - \tau_3(t) + A_2(\tau_3(t)), \quad (40)$$

where $A_2(\tau_3(t))$ is the age of node 2 at the time of the last successful transmission over link e_3 , namely $\tau_3(t)$, and $t - \tau_3(t)$ is the time elapsed since then. Substituting (39) in (40), we obtain

$$A_3(t) = t - \tau_3(t) + [\tau_3(t) - \tau_2(\tau_3(t))], \quad (41)$$

$$= t - \tau_2(\tau_3(t)). \quad (42)$$

Iterating this over j links, we get

$$A_j(t) = t - \tau_2(\tau_3(\dots \tau_j(t) \dots)), \quad (43)$$

for all t and nodes $j \in \{2, 3, \dots, N\}$. Taking expectation, we get

$$\mathbb{E}[A_j(t)] = t - \mathbb{E}[\tau_2(\tau_3(\dots \tau_j(t) \dots))], \quad (44)$$

$$= \sum_{i=2}^j \frac{1}{\alpha_i}, \quad (45)$$

where the last equality follows from the following Lemma 5, and substituting $j = N$ we obtain (35), and also, the result.

Lemma 5: $\tau_j(t)$ for $j \in \{2, 3, \dots, N\}$ satisfy

$$\mathbb{E}[\tau_2(\tau_3(\dots \tau_j(t) \dots))] = t - \sum_{i=2}^j \frac{1}{\alpha_i}. \quad (46)$$

Proof: From (36), we know that

$$\tau_i(t) \stackrel{d}{=} t - X_{e_i}, \quad (47)$$

where X_{e_i} is a geometrically distributed random variable given by (37), and therefore,

$$\mathbb{E}[\tau_i(t)] = \mathbb{E}[t - X_{e_i}] = t - \frac{1}{\alpha_i}, \quad (48)$$

for all $i \in \{2, 3, \dots, N\}$. Now from (47), we can obtain

$$\tau_i(\tau_{i+1}(t)) \stackrel{d}{=} \tau_{i+1}(t) - X_{e_i}, \quad (49)$$

and therefore,

$$\mathbb{E}[\tau_i(\tau_{i+1}(t))] = \mathbb{E}[\tau_{i+1}(t)] - \mathbb{E}[X_{e_i}], \quad (50)$$

$$= t - \frac{1}{\alpha_{i+1}} - \frac{1}{\alpha_i}, \quad (51)$$

where the last equality follows from (48) and (38). Iterating this u times we obtain:

$$\mathbb{E}[\tau_i(\tau_{i+1}(\dots \tau_{i+u}(t) \dots))] = t - \sum_{l=i}^{i+u} \frac{1}{\alpha_l}. \quad (52)$$

Substituting $i = 2$ and $i + u = j$ we get the result. ■

C. Proof of Lemma 1

The optimization problem (10) is given by

$$\begin{aligned} & \text{Minimize} \sum_{\mathbf{x} \in \mathbb{X}} \sum_{k \in [K]} \sum_{e \in p^k} \frac{w^k}{\gamma_e f_e^k}, \\ & \text{subject to } \mathbf{f} = M\mathbf{x}. \end{aligned} \quad (53)$$

Consider the set of tuples $H = \{(k, e), \forall k \in [K], \forall e \in p^k\}$. The objective in (10) involves one term for each element in H . To create a new single-hop network with the same age minimization problem, we create a source-destination pair corresponding to each element $h = (k, e) \in H$. The link reliability of the edge for the source-destination pair h is given by γ_e , while the weight for AoI at the destination is given by w^k . The set of feasible activations for the original multihop network $\mathcal{A}$ contains interference free choices of flow and edge activations of the form $\{(k_1, e_1), (k_2, e_2), \dots\}$. For our new multihop network, we simply translate this to general interference constraints. For example, if $\{(k_1, e_1), (k_2, e_2)\} \in \mathcal{A}$, then the sources corresponding to (k_1, e_1) and (k_2, e_2) can attempt to transmit simultaneously in the new single-hop network without interference. Now, we can construct the single-hop age minimization problem over this new network using (34) as below:

$$\begin{aligned} & \text{Minimize} \sum_{\mathbf{x} \in \mathbb{X}} \sum_{(k, e) \in H} \frac{w^k}{\gamma_e f_e^k}, \\ & \text{subject to } \mathbf{f} = M\mathbf{x}. \end{aligned} \quad (54)$$

Note that this is identical to (10), which completes the proof.

D. Proof of Lemma 2

The per time-slot cost in A_{ave} (in (14)) is

$$C(t) = \sum_{k=1}^K \sum_{j \in C_k \cup D_k} w_j^k A_j^k(t). \quad (55)$$

Note that the age evolution in (1) can be re-written as

$$\begin{aligned} A_j^k(t+1) &= 1 + A_j^k(t) \\ &\quad - \sum_i U_{ij}^k(t) S_{ij}(t) [A_j^k(t) - A_i^k(t)]^+, \end{aligned} \quad (56)$$

for all $j \in C_k \cup D_k$, $i \in \{k\} \cup C_k$ that are neighbors of j ; in (56) $[x]^+$ is used to denotes $\max\{x, 0\}$.

From (55)-(56), we deduce that the per time-slot cost difference will be

$$\begin{aligned} & C(t+1) - C(t) \\ &= \sum_{k=1}^K \sum_{j \in C_k \cup D_k} w_j^k \\ &\quad - \sum_{(i,j) \in \bar{E}} \sum_{k=1}^K U_{ij}^k(t) S_{ij}(t) [A_j^k(t) - A_i^k(t)]^+, \end{aligned} \quad (57)$$

where $\bar{E}$ that are present in the subgraph induced by all the source, commissioned, and destination nodes. $\bar{E}$ accounts for

all the links on which updates will be forwarded for at least one flow. Writing (57) in terms of weights $\Delta_{ij}^k(t)$, we have:

$$C(t+1) - C(t) = \sum_{k=1}^K \sum_{j \in C_k \cup D_k} w_j^k - \sum_{(i,j) \in \bar{E}} \sum_{k=1}^K \Delta_{ij}^k(t). \quad (58)$$

The age-difference policy maximizes the sum

$$\sum_{(i,j) \in \bar{E}} \sum_{k=1}^K \Delta_{ij}^k(t). \quad (59)$$

As a result, it minimizes $C(t+1)$, given all occurrences till time t . This shows that the age-difference policy is a myopic policy for the average age defined in (14).

E. Proof of Lemma 3

We will prove this under the assumption that the AoI cost functions $g_j^k(\cdot)$ are upper-bounded by a fixed constant D for every source-destination pair (k, j) . This is a mild assumption because D can be set to a very high value (in the order of years) which will never be attained in practical systems under any reasonable policy.

We note that the arrival process to the debt queue $Q_j^k(t)$ is given by the effective age process $B_j^k(t)$, while the departures in every time-slot are just α_j^k . Using the boundedness assumption, both arrivals and departures are strictly upper-bounded by D . The result immediately follows from Theorem 2(c) in [34] which relates mean-rate stability of a queue to time-averages of the arrival and departure processes.

F. Proof of Lemma 1

The debt queues in this setting evolve as follows:

$$Q_i(t+1) = \left[Q_i(t) + g_i(A_i(t+1)) - \alpha_i \right]^+, \forall i. \quad (60)$$

The AoI evolves as:

$$A_i(t+1) \begin{cases} A_i(t) + 1, & \text{if } U_{e_i}(t) S_{e_i}(t) = 0 \\ 1, & \text{if } U_{e_i}(t) S_{e_i}(t) = 1. \end{cases} \quad (61)$$

Here $S_{e_i}(t) = 1$ i.i.d. with probability γ_i in every time-slot.

Let $\Delta(t) \triangleq L(t+1) - L(t)$. Then,

$$\begin{aligned} \mathbb{E}[\Delta(t)] &= \sum_i \mathbb{E} \left[(Q_i(t+1))^2 - (Q_i(t))^2 \right] \\ &\leq \sum_i \mathbb{E} \left[\alpha_i^2 - 2\alpha_i Q_i(t) + (g_i(A_i(t+1)))^2 \right. \\ &\quad \left. + 2Q_i(t)g_i(A_i(t+1)) - 2\alpha_i g_i(A_i(t+1)) \right] \\ &\leq \sum_i \left[D^2 + 2Q_i(t)(\mathbb{E}[g_i(A_i(t+1))]) - \alpha_i \right] \end{aligned} \quad (62)$$

The first inequality follows from the evolution of debt queues. The second inequality follows from the boundedness assumption on $g_i(\cdot)$, i.e. $g_i(h) \leq D, \forall h$. Now, we will

minimize the RHS of the expression above. We can drop the term D^2 since it is a constant.

$$\begin{aligned}
& \argmin_{\pi(t) \in 1, \dots, N-1} \sum_i Q_i(t) (\mathbb{E}[g_i(A_i(t+1))] - \alpha_i) \\
&= \argmin_{\pi(t) \in 1, \dots, N-1} \sum_i Q_i(t) \mathbb{E}[g_i(A_i(t+1))] \\
&= \argmin_{j \in 1, \dots, N-1} \left[\sum_i \left(Q_i(t) g_i(A_i(t) + 1) \right) \right. \\
&\quad \left. + \gamma_j Q_j(t) (g_j(1) - g_j(A_j(t) + 1)) \right] \\
&= \argmax_{j \in 1, \dots, N-1} \left[\gamma_j Q_j(t) (g_j(A_j(t) + 1) - g_j(1)) \right] \quad (63)
\end{aligned}$$

The first equality follows since $Q_i(t)\alpha_i$ does not depend on the scheduling decision $\pi(t)$. The second equality follows from the evolution of AoI given $\pi(t) = j$. The third equality follows since the summation term does not depend on the scheduling choice j . This completes the proof.

ACKNOWLEDGMENT

The authors also thank Shahab Farazi for sharing with them an implementation of the MCDS scheme from [28].

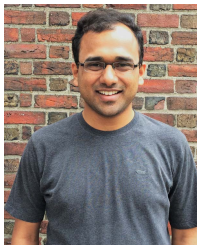
REFERENCES

- [1] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2731–2735.
- [2] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Proc. 8th Annu. IEEE Conf. Sensor, Mesh Ad Hoc Commun. Netw. (SECON)*, Jun. 2011, pp. 350–358.
- [3] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in *Proc. IEEE ISIT*, Apr. 2015, pp. 1681–1685.
- [4] S. K. Kaul, R. D. Yates, and M. Gruteser, "Status updates through queues," in *Proc. 46th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2012, pp. 1–6.
- [5] E. Najm and R. Nasser, "Age of information: The gamma awakening," in *Proc. IEEE ISIT*, Jul. 2016, pp. 2574–2578.
- [6] M. Costa, M. Codreanu, and A. Ephremides, "Age of information with packet management," in *Proc. IEEE ISIT*, Jun. 2014, pp. 1583–1587.
- [7] C. Kam, S. Kompella, and A. Ephremides, "Age of information under random updates," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 66–70.
- [8] C. Kam, S. Kompella, and A. Ephremides, "Effect of message transmission diversity on status age," in *Proc. IEEE ISIT*, Jun. 2014, pp. 2411–2415.
- [9] K. Chen and L. Huang, "Age-of-information in the presence of error," in *Proc. IEEE ISIT*, Jul. 2016, pp. 2579–2583.
- [10] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka, "A general formula for the stationary distribution of the age of information and its application to single-server queues," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 8305–8324, Aug. 2019.
- [11] I. Kadota, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Minimizing the age of information in broadcast wireless networks," in *Proc. Allerton Conf.*, Sep. 2016, pp. 844–851.
- [12] Y.-P. Hsu, E. Modiano, and L. Duan, "Age of information: Design and analysis of optimal scheduling algorithms," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 1–5.
- [13] R. Talak, I. Kadota, S. Karaman, and E. Modiano, "Scheduling policies for age minimization in wireless networks with unknown channel state," in *Proc. ISIT*, Jun. 2018, pp. 2564–2568.
- [14] I. Kadota, A. Sinha, and E. Modiano, "Optimizing age of information in wireless networks with throughput constraints," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 1844–1852.
- [15] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2637–2650, Dec. 2018.
- [16] R. Talak, S. Karaman, and E. Modiano, "Optimizing information freshness in wireless networks under general interference constraints," *IEEE/ACM Trans. Netw.*, vol. 28, no. 1, pp. 15–28, Feb. 2019.
- [17] R. Talak, S. Karaman, and E. Modiano, "Optimizing age of information in wireless networks with perfect channel state information," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1765–1778, Aug. 2020.
- [18] A. Kosta, N. Pappas, and V. Angelakis, "Age of information: A new concept, metric, and tool," *Found. Trends Netw.*, vol. 12, no. 3, pp. 162–259, 2017.
- [19] Y. Sun, I. Kadota, R. Talak, and E. Modiano, "Age of information: A new metric for information freshness," *Synth. Lectures Commun. Netw.*, vol. 12, no. 2, pp. 1–224, Dec. 2019.
- [20] S. K. Kaul and R. Yates, "Status updates over unreliable multiaccess channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 331–335.
- [21] O. T. Yavascan and E. Uysal, "Analysis of slotted ALOHA with an age threshold," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1456–1470, May 2021.
- [22] I. Kadota and E. Modiano, "Age of information in random access networks with stochastic arrivals," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.
- [23] P. R. Jhunjhunwala and S. Moharir, "Age-of-information aware scheduling," in *Proc. Int. Conf. Signal Process. Commun. (SPCOM)*, Jul. 2018, pp. 222–226.
- [24] V. Tripathi and E. Modiano, "A whittle index approach to minimizing functions of age of information," in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2019, pp. 1160–1167.
- [25] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Age-optimal information updates in multihop networks," in *Proc. IEEE ISIT*, Jun. 2017, pp. 576–580.
- [26] B. Buyukates, A. Soysal, and S. Ulukus, "Age of information in multihop multicast networks," *J. Commun. Netw.*, vol. 21, no. 3, pp. 256–267, 2019.
- [27] Q. Liu, H. Zeng, and M. Chen, "Minimizing age-of-information with throughput requirements in multi-path network communication," in *Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jul. 2019, pp. 41–50.
- [28] S. Farazi, A. G. Klein, J. A. McNeill, and D. R. Brown, "On the age of information in multi-source multi-hop wireless status update networks," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2018, pp. 1–5.
- [29] S. Farazi, A. G. Klein, and D. R. Brown, "Fundamental bounds on the age of information in general multi-hop interference networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2019, pp. 96–101.
- [30] R. Talak, S. Karaman, and E. Modiano, "Minimizing age-of-information in multi-hop wireless networks," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput.*, Oct. 2017, pp. 486–493.
- [31] V. Tripathi and E. Modiano, "Age debt: A general framework for minimizing age of information," in *Proc. IEEE Conf. Comput. Commun. Workshops*, May 2021, pp. 1–6.
- [32] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Minimizing the age of information through queues," *IEEE Trans. Inf. Theory*, vol. 65, no. 8, pp. 5215–5232, Aug. 2019.
- [33] I.-H. Hou, V. Borkar, and P. R. Kumar, "A theory of QoS for wireless," in *Proc. 28th Conf. Comput. Commun.*, Apr. 2009, pp. 486–494.
- [34] M. J. Neely, "Stability and capacity regions or discrete time queueing networks," 2010, *arXiv:1003.3396*.
- [35] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–44, 2006.



Vishrant Tripathi (Graduate Student Member, IEEE) received the Bachelor of Technology (B.Tech.) degree from the Department of Electrical Engineering, Indian Institute of Technology, Bombay (IIT-B), India, in 2017. He is currently pursuing the Ph.D. degree with the Laboratory of Information and Decision Systems (LIDS), Massachusetts Institute of Technology (MIT).

His research interests include modeling, analysis and design of communication networks, with emphasis on wireless and real-time networks. His current focus is on scheduling problems in networked control systems, robotics and federated learning.



Rajat Talak received the B.Tech. degree from the Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, India, in 2010, the Master of Science degree from the Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru, India, in 2013, and the Ph.D. degree from the Laboratory of Information and Decision Systems, Massachusetts Institute of Technology (MIT) in 2020.

He is currently a Research Scientist with the Department of Aeronautics and Astronautics, MIT. Prior to this, he was a Post-Doctoral Associate with the Department of Aeronautics and Astronautics, MIT. His research interests include the field of networked autonomy, robotic perception, certifiable models, and high-level 3D scene understanding. His work includes seminal results on guaranteeing information freshness in single and multi-hop communication network of autonomous agents; providing a theoretical framework for set-based uncertainty models on graphs; developing provably expressive graph neural network architecture for 3D scene graphs; and a novel self-supervised learning procedure that uses certification in the training loop. He was a recipient of ACM MobiHoc 2018 Best Paper Award and Prof. F. M. Mowdawalla medal for his master's thesis.



Eytan Modiano (Fellow, IEEE) received the B.S. degree in electrical engineering and computer science from the University of Connecticut at Storrs in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, USA, in 1989 and 1992, respectively.

He is currently a Richard C. Maclaurin Professor with the Laboratory for Information and Decision Systems (LIDS), Department of Aeronautics and Astronautics, MIT. Prior to joining the Faculty at MIT in 1999, he was a Naval Research Laboratory Fellow from 1987 to 1992, a National Research Council Post-Doctoral Fellow from 1992 to 1993, and a member of the Technical Staff at the MIT Lincoln Laboratory from 1993 to 1999. His research interests include modeling, analysis and design of communication networks and protocols. He had served on the IEEE Fellow Committee in 2014 and 2015, and an Associate Fellow for the AIAA. He received the Infocom Achievement Award in 2020 for contributions to the analysis and design of cross-layer resource allocation algorithms for wireless, optical, and satellite networks. He was a co-recipient of the Infocom 2018 Best Paper Award, the MobiHoc 2018 Best Paper Award, the MobiHoc 2016 Best Paper Award, the Wiopt 2013 Best Paper Award, and the Sigmetrics 2006 Best Paper Award. He was the Technical Program Co-Chair of IEEE Wiopt 2006, IEEE Infocom 2007, ACM MobiHoc 2007, and DRCN 2015; and a General Co-Chair of Wiopt 2021. He was the Editor-in-Chief of IEEE/ACM TRANSACTIONS ON NETWORKING (2017–2020), and served as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY and IEEE/ACM TRANSACTIONS ON NETWORKING.