Fresh-CSMA: A Distributed Protocol for Minimizing Age of Information

Vishrant Tripathi, Nicholas Jones, and Eytan Modiano

Abstract—**We consider the design of distributed scheduling algorithms that minimize age of information (AoI) in single-hop wireless networks. The centralized max-weight policy is known to be nearly optimal in this setting; hence, our goal is to design a distributed carrier sense multiple access (CSMA) scheme that can mimic its performance. To that end, we propose a distributed protocol called Fresh-CSMA and show that in an idealized setting, Fresh-CSMA can match the scheduling decisions of the max-weight policy with high probability in each frame, and also match the theoretical performance guarantees of the max-weight policy over the entire time horizon. We then consider a more realistic setting and study the impact of protocol parameters on the probability of collisions and the overhead caused by the distributed nature of the protocol. We also consider the monitoring of Markov sources and extend our approach to CSMA protocols that incorporate age of incorrect information (AoII) instead of AoI. Finally, we provide simulations that support our theoretical results and show that the performance gap between the ideal and realistic versions of Fresh-CSMA is small.**

Index Terms—**Age of information, distributed protocols, scheduling, wireless networks.**

I. INTRODUCTION

MANY emerging applications require timely delivery of information updates over communication networks. Age of information (AoI) is a metric that captures this notion of timeliness of received information at a destination [1]–[3]. Unlike packet delay, AoI measures the lag in obtaining information at a destination node, and is therefore suited for applications involving time sensitive updates. Age of information, at a destination, is defined as the time that has elapsed since the last received information update was generated at the source. AoI, upon reception of a new update, drops to the time that has elapsed since generation of the update, and grows linearly otherwise.

The design of scheduling policies to minimize age of information metrics over single-hop wireless networks has received special interest over the past few years. In [4], [5], the authors consider a single-hop broadcast setting with the aim of minimizing weighted sum AoI. In [6], the authors also study the minimization of the weighted sum of AoI but under general interference constraints. These lines of work prove constant factor optimality of three different classes of scheduling policies; stationary randomized, max-weight and Whittle index.

In [7], the authors prove asymptotic optimality of the Whittle index policy and in [8], the authors extend prior results to different sampling behaviors, update sizes and transmission times in a single-hop broadcast setting. In [9], [10], the authors extend the AoI minimization framework to consider general nonlinear cost functions of AoI.

Importantly, all of the works above focus on the design of *centralized scheduling algorithms*. Specifically, at the beginning of each time-slot, the base station looks at the AoI values for each node in the network and then decides which node to poll for an update. This requires support for polling protocols, which might not be available at the MAC layer and might involve excessive overhead for networks with many nodes. This has motivated the need to study distributed schemes for information freshness in wireless networks.

In [11], the authors consider a simple class of distributed algorithms - each node transmits using a fixed attempt probability in each time-slot, and they design a scheme to find the attempt probabilities that minimize weighted sum AoI. In [12], the authors consider a single-hop setting with stochastic arrivals and solve the AoI minimization problem by deriving a Whittle index and propose a heuristic ALOHA-like scheme called index-prioritized-random-access (IPRA) where a node is active with a fixed probability but only when its AoI exceeds a specified threshold. The idea of ALOHA with thresholds has been explored in further detail in subsequent works. In [13], [14], the authors study the performance of ALOHA style random access protocols for information freshness and propose the idea of "thinning", where only nodes with AoI greater than a specified threshold remain active. Along similar lines, the performance of threshold-ALOHA for AoI minimization is also analyzed in [15], [16] with performance bounds derived in a symmetric setting.

Another class of distributed protocols commonly used in wireless networks for medium access is carrier sense multiple access with collision avoidance, also known as CSMA/CA. Throughout this work, when we use the term CSMA, we use it to denote CSMA/CA style protocols. Age of information has also been analyzed in settings where nodes employ CSMA [13], [17]–[21]. In [17], the authors analyze an idealized version of IEEE 802.11 CSMA and optimize the backoff timer parameters to minimize AoI. They show that this version of CSMA has poor delay and freshness performance in certain settings and suggest the need for new distributed scheduling schemes for AoI. In [18], [19], the authors analyze AoI under standard CSMA in broadcast environments. In [20], [21], the authors study sleep-wake carrier sensing based scheduling with the goal of minimizing energy consumption together with AoI.
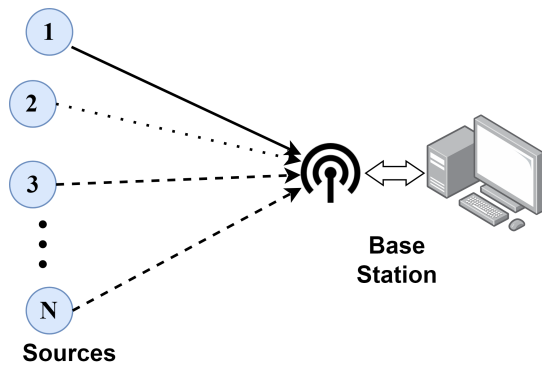
Fig. 1. Single-hop broadcast network with N sources sending updates to a base station over a shared channel.

The CSMA protocol has been well studied in wireless networks for a long time, especially for optimizing throughput and utility. It was shown in [22] that CSMA tends to outperform ALOHA in terms of both throughput and delay. In [23], the author developed an approximation that allows closed-form analysis for the IEEE 802.11 implementation of CSMA. More recently, there has been work on throughput and utility optimization by trying to replicate centralized scheduling policies' behavior using CSMA style schemes [24]–[27]. Typically, these works involve modifying the way CSMA backoff timers work by adding dependence on the current network state (e.g., queue lengths) and then analyzing performance guarantees by comparing to a centralized scheduling scheme. Our analysis and approach are motivated by this line of work, in particular the Fast-CSMA protocol proposed in [27].

In this work, we propose Fresh-CSMA to replicate the behavior of centralized scheduling schemes that minimize AoI. In Section II we discuss our system model and set up the single-hop weighted age minimization problem. In Section III we introduce the Fresh-CSMA protocol in an idealized setting and provide performance guarantees that show that it can closely match the centralized max-weight scheduling policy both per time-slot and over the entire time horizon. In Section IV, we relax some of the assumptions from our idealized model and study the Fresh-CSMA protocol under a more realistic setting. We analyze two keys aspects - the probability of collision and the total time lost due to the backoff timers during which the channel remains idle. In Section V, we consider the recently proposed information freshness metric called age of incorrect information (AoII) and extend our CSMA design to incorporate this metric. In Section VI, we provide simulations that support our theoretical results.

A preliminary version of our paper appeared in the conference proceedings of IEEE INFOCOM 2023 [28].

II. SYSTEM MODEL

Consider a single-hop wireless network with N sources generating real-time status updates that need to be sent to a monitoring base station (see Fig. 1). We consider a slotted system in which each source takes a single time-slot to transmit an update to the base station. Due to interference, only one of the sources can transmit successfully in any given time-slot. If multiple sources decide to transmit, a collision occurs and the transmitted updates are lost. We assume reliable delivery of updates in the absence of collisions.

For every source i, the age of information at the base station $A_i(t)$ measures the time elapsed since it received a fresh information update from the source. We assume active sources, i.e., in any time-slot, sources can generate fresh updates at will. Let $s(t)$ be the set of sources transmitting in time-slot t. Then, the age of information $A_i(t)$ evolves as:

$$A_i(t+1) = \begin{cases} 1, & \text{if } s(t) = \{i\} \\ A_i(t) + 1, & \text{otherwise.} \end{cases} \quad (1)$$

The metric of interest in this work will be average AoI, which is simply the long-term time-average of the AoI process. Specifically,

$$\bar{A}_i \triangleq \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} A_i(t). \quad (2)$$

The goal of this work is to design a *distributed* wireless scheduling policy that minimizes the weighted sum of average AoI across all sources:

$$\underset{\pi}{\operatorname{argmin}} \left(\limsup_{T \to \infty} \left[\frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} w_i A_i(t) \right] \right). \quad (3)$$

Here, the weights $\{w_1, w_2, \cdots, w_N\}$ are positive *integers* that denote the relative importance of each source to the overall monitoring or control application.

III. DISTRIBUTED SCHEDULING DESIGN

Before we introduce our distributed scheduling design, we briefly discuss key results from prior works that have looked at the same problem but from a centralized perspective.

In [4], the authors considered a similar single-hop network setting with the goal of minimizing weighted-sum average AoI, i.e., solving (3). First, they considered the class of stationary randomized policies. Each policy within this class is simply a probability distribution over the set of sources and the scheduling decision is sampled from this distribution i.i.d. at the beginning of every time-slot. They showed that under the optimal stationary randomized policies that solve (3), each source i is scheduled with probability

$$\pi_i^* = \frac{\sqrt{w_i}}{\sum_{i=1}^{N} \sqrt{w_i}}, \forall i \in [N]. \quad (4)$$

They further showed that the best stationary randomized policies can be at most a factor of two away from the overall optimal policy.

They also proposed a centralized policy motivated by Lyapunov drift arguments called the max-weight policy. This policy, under reliable channels, makes scheduling decisions $\pi^{mw}(t)$ as follows:

$$\pi^{mw}(t) = \underset{j \in [N]}{\operatorname{argmax}} \left(w_j A_j^2(t) \right). \quad (5)$$

In [29], the authors showed that this policy is at most a factor of two away from optimal using Lyapunov drift arguments.

However, unlike stationary randomized policies, this policy turns out to be *nearly optimal* in practice.

The goal of our distributed design is to replicate the decision making and performance guarantees of max-weight policies of the form (5). Throughout this work, we will benchmark the performance of distributed protocols in comparison to an idealized implementation of the max-weight policy that has no polling overhead.

A. Fresh-CSMA

In general, a CSMA/CA style protocol involves the following steps. First, a node senses the channel to see if it is free. If the node determines the channel to be available, it starts transmitting a packet. Otherwise, if the channel is occupied, it generates a random backoff time and starts a timer counting down from this value. During this period, the node continuously senses the channel and only counts down *when the channel is free*. Once the timer hits zero, the node transmits a frame. Note that the timer can only hit zero when the channel is known to be free. Depending on whether an ACK is received or not from the receiving station, the node updates its random backoff timer parameters for the next transmission.

The details of how to sample the backoff times, how to update them in case of a collision or re-transmission, and how to implement channel sensing determine the exact flavor of CSMA being implemented. To develop our scheme and provide tractable analysis, we will consider an idealized channel sensing setup that is commonly used in theoretical works addressing CSMA [24]–[27]. This involves making the following key assumptions.

Assumption *1:* Backoff timers are implemented in continuous time.

Assumption *2:* Carrier sensing happens instantly.

Assumption *3:* There is a discrete slotted system and all nodes start their backoff timers at the beginning of each time-slot.

Assumption *4:* Backoff timers are implemented with arbitrary precision, and can be made negligible in comparison to the duration of a time-slot.

Under these assumptions, a version of the classic CSMA protocol that uses exponential backoff timers is described in Algorithm 1. We use t to denote the discrete time-slots and τ to denote continuous time within each time-slot. We normalize the time-slot length to be 1 so τ is a continuous timer that increases from 0 to 1 within each time-slot.

The protocol consists of two key ideas. First, each source i generates a random timer $Z_i(t)$ that is i.i.d. exponentially distributed. Second, the source with the timer that runs out first gets to transmit in the entire time-slot t, i.e.,

$$\pi(t) = \underset{i \in [N]}{\operatorname{argmin}}\ Z_i(t). \tag{6}$$

As a consequence of Assumptions 1–3, note that a packet collision happens only if two nodes choose the exact same backoff times. Since the probability that two exponential

---

Algorithm 1: Idealized CSMA

Input : parameter $\alpha > 1$
1 **while** $t \in 1, \cdots, T$ **do**
2 　**At every source** $i \in 1, \cdots, N$ **do**
3 　　Generate a random timer $Z_i(t) \sim \exp(\alpha)$.
4 　　**while** $\tau < Z_i(t)$ **do**
5 　　　Stay silent
6 　　**end**
7 　　**if** *Channel is free* **then**
8 　　　Transmit
9 　　**end**
10 　
11 **end**

Algorithm 2: Idealized Fresh-CSMA

Input : parameters $\alpha > 1, \zeta \ll 1$
1 **while** $t \in 1, \cdots, T$ **do**
2 　**At every source** $i \in 1, \cdots, N$ **do**
3 　　Generate a random timer
　　　$Z_i(t) \sim \exp\left(\alpha^{w_i A_i^2(t)}\right).$
4 　　**while** $\tau < \zeta Z_i(t)$ **do**
5 　　　Stay silent
6 　　**end**
7 　　**if** *Channel is free* **then**
8 　　　Transmit
9 　　**end**
10 　
11 **end**

---

random variables take the exact same value is zero, so the probability of packet collisions is also zero in this idealized setup. Further, due to Assumption 4, we can scale the backoff timers by a factor ζ such that they are negligible in comparison to the slot length, i.e., $\zeta Z_i(t) \ll 1, \forall i$ with high probability. Thus, the procedure of figuring out which node has the smallest backoff period takes negligible time compared to the length of the time-slot, allowing us to ignore the backoff overhead of the random access scheme.

Next, we modify this CSMA protocol to create Fresh-CSMA, described in Algorithm 2. This is also an idealized distributed protocol, but with the goal of replicating the behavior of the max-weight scheduling policy (5) for AoI minimization. This style of CSMA is motivated by the fast-CSMA protocol proposed in [27].

Note that Fresh-CSMA consists of two key steps. First, each source i generates a random timer $\zeta Z_i(t)$ where $Z_i(t)$ is exponentially distributed with the parameter $\alpha^{w_i A_i^2(t)}$. Then, the source with the timer that runs out first gets to transmit in the entire time-slot t, i.e., according to (6).

Importantly, each source only requires knowledge of its own AoI and scheduling weight to compute the backoff timer, thus maintaining the distributed nature of the protocol. The following lemma describes the structure of scheduling decisions made by this scheduling scheme.

Lemma *1:* For Fresh-CSMA at time-slot t, the probability that source i is scheduled is given by:

$$r_i(t) = \frac{\alpha^{w_i A_i^2(t)}}{\sum\limits_{j=1}^{N} \alpha^{w_j A_j^2(t)}}. \tag{7}$$

Proof: First, note that the scheduling decision at time-slot t is given by (6) where $Z_i(t)$ are independent and exponentially distributed with the parameters $\alpha^{w_i A_i^2(t)}$. Let $Z(t) \triangleq \min\limits_{i \in [N]} Z_i(t)$. Then, $Z(t)$ is the minimum of N independent exponential random variables. Thus, it is also exponentially distributed and with the parameter $\sum_{j=1}^{N} \alpha^{w_j A_j^2(t)}$. Let $\lambda_i \triangleq \alpha^{w_i A_i^2(t)}$. We are interested in calculating the probability

$$\begin{aligned}
r_i(t) = \mathbb{P}\big(\pi(t) = i\big) &= \mathbb{P}\big(Z(t) = Z_i(t)\big) \\
&= \int_0^\infty f_{Z_i(t)}(x) \prod_{k \neq i} \mathbb{P}\big(Z_k(t) > x\big) dx \\
&= \int_0^\infty \lambda_i e^{-\lambda_i x} \prod_{k \neq i} e^{-\lambda_k x} dx = \frac{\lambda_i}{\sum_{k=1}^{N} \lambda_k}.
\end{aligned} \tag{8}$$

This completes the proof. ∎

Using Lemma 1, we next show that if the parameter α is set to be large enough, then in any particular time-slot, Fresh-CSMA will make the same scheduling decision as the max-weight policy with high probability.

Theorem *1:* Given any $\delta \in (0, 1)$ and $A_1(t), \cdots, A_N(t)$; if we set $\alpha \geq (N-1)\frac{1-\delta}{\delta}$, then the following holds

$$\mathbb{P}\left(\pi^{Fresh-CSMA}(t) = \pi^{mw}(t)\right) \geq 1 - \delta. \tag{9}$$

Here, $\pi^{mw}(t)$ is the max-weight scheduling decision given by (5), while $\pi^{Fresh-CSMA}(t)$ is the scheduling decision made by Fresh-CSMA.

Proof: We divide the proof into two parts.

Case 1: The expression $\underset{j \in [N]}{\operatorname{argmax}}\left(w_j A_j^2(t)\right)$ has a unique maximum. Let this maximum be the source 1 without loss of generality. Then, the max-weight decision is to schedule source 1. Since we have assumed all the weights w_i to be positive integers and the AoIs $A_i(t)$ are integers by definition, so the quantities $w_i A_i^2(t)$ are also positive integers and the following must hold:

$$w_1 A_1^2(t) - w_i A_i^2(t) \geq 1, \forall i \neq 1. \tag{10}$$

Note that in the special case of $w_i = 1, \forall i$, (10) holds when the AoIs across the different sources are unique.

Now, applying Lemma 1, we can calculate the following probability

$$\begin{aligned}
\mathbb{P}\left(\pi^{Fresh-CSMA}(t) = 1\right) &= \frac{\alpha^{w_1 A_1^2(t)}}{\sum\limits_{j=1}^{N} \alpha^{w_j A_j^2(t)}} \\
&= \frac{1}{1 + \sum_{i=2}^{N} \alpha^{w_i A_i^2(t) - w_1 A_1^2(t)}} \\
&\geq \frac{1}{1 + (N-1)\alpha^{-1}} \\
&\geq 1 - \delta.
\end{aligned} \tag{11}$$

The first inequality follows by using (10) while the second inequality follows due to the fact that $\alpha \geq (N-1)(1-\delta)/\delta$.

Case 2: The expression $\underset{j \in [N]}{\operatorname{argmax}}\left(w_j A_j^2(t)\right)$ has multiple maxima. Suppose that the set of maxima is given by the nodes $\{1, \cdots, k\}$. Then, the max-weight policy will choose one of these sources to be scheduled. We want to lower bound the probability that Fresh-CSMA chooses a node from within this set. To do so, we first make a similar observation as in the case above.

$$w_1 A_1^2(t) - w_j A_j^2(t) \geq 1, \forall j = k+1, \cdots, N \tag{12}$$

Using Lemma 1, we calculate the probability of interest

$$\begin{aligned}
\mathbb{P}\left(\pi^{Fresh-CSMA}(t) \in \{1, \cdots, k\}\right) &= \frac{k\alpha^{w_1 A_1^2(t)}}{\sum\limits_{j=1}^{N} \alpha^{w_j A_j^2(t)}} \\
&= \frac{k}{k + \sum_{i=k+1}^{N} \alpha^{w_i A_i^2(t) - w_1 A_1^2(t)}} \\
&\geq \frac{1}{1 + (N-k)\alpha^{-1}} \\
&\geq 1 - \delta.
\end{aligned} \tag{13}$$

As before, the first inequality follows by using (12) while the second inequality follows due to the fact that $\alpha \geq (N-1)(1-\delta)/\delta$. This completes the proof. ∎

Next, we show that our idealized Fresh-CSMA protocol has the same theoretical long-term performance guarantees as the max-weight policy.

Theorem *2:* Given any set of integer weights $w_1, \cdots, w_N$, if we set $\alpha > \left(\frac{(N-1)\sum_{i=1}^{N} \sqrt{w_i}}{\min_j \sqrt{w_j}}\right)$, then the following holds:

$$\frac{\sum_{i=1}^{N} w_i \bar{A}_i^{csma}}{\sum_{i=1}^{N} w_i \bar{A}_i^{opt}} \leq 2. \tag{14}$$

Here, $\bar{A}_i^{csma}$ is the average AoI for source i under the Fresh-CSMA policy while $\bar{A}_i^{opt}$ is the average AoI of source i under an optimal policy π^{opt} that solves the age minimization problem (3).

Proof: Consider the linear Lyapunov function as defined below:

$$L(t) \triangleq \sum_{i=1}^{N} \sqrt{w_i} A_i(t). \tag{15}$$

We can define the one-slot Lyapunov drift $\Delta(t) \triangleq L(t+1) - L(t)$. The main challenge in proving the performance bound above, is to first show an intermediate result relating the Lyapunov drift of the Fresh-CSMA policy to that of the optimal stationary randomized policy described by (4).

Lemma *2:* Consider any $\boldsymbol{A(t)} = \{A_1, \cdots, A_N(t)\}$. Let $\Delta^{csma}(t)$ be the one-slot Lyapunov drift of the Fresh-CSMA policy and $\Delta^{sr}(t)$ be the one-slot Lyapunov drift of the optimal stationary randomized policy. Then, the following holds:

$$\mathbb{E}\left[\Delta^{csma}(t)\Big|\boldsymbol{A(t)}\right] \le \mathbb{E}\left[\Delta^{sr}(t)\Big|\boldsymbol{A(t)}\right], \forall \boldsymbol{A(t)}. \tag{16}$$

Proof: We first calculate an expression for the drift of the Fresh-CSMA policy. Recall that $r_j(t) \triangleq \mathbb{P}\left(\pi^{AoI-CMSA}(t) = j\right)$ and is given by (7).

$$\mathbb{E}\left[\Delta^{csma}(t)\Big|\boldsymbol{A(t)}\right]$$
$$= \sum_{j=1}^{N} r_j(t)\left(\sqrt{w_j} + \sum_{i \ne j} \sqrt{w_i}(A_i(t)+1)\right) - \sum_{j=1}^{N} \sqrt{w_j} A_j(t)$$
$$= \sum_{j=1}^{N} \sqrt{w_j} - \sum_{j=1}^{N} r_j(t)\sqrt{w_j} A_j(t). \tag{17}$$

Repeating the above steps for the optimal stationary randomized policy, we get:

$$\mathbb{E}\left[\Delta^{sr}(t)\Big|\boldsymbol{A(t)}\right] = \sum_{j=1}^{N} \sqrt{w_j} - \sum_{j=1}^{N} \pi_j^* \sqrt{w_j} A_j(t). \tag{18}$$

Recall that π_j^* are scheduling probabilities for the optimal stationary randomized policy, given by (4).

Consider the difference between (17) and (18)

$$\mathbb{E}\left[\Delta^{csma}(t)\Big|\boldsymbol{A(t)}\right] - \mathbb{E}\left[\Delta^{sr}(t)\Big|\boldsymbol{A(t)}\right]$$
$$= \sum_{j=1}^{N} \sqrt{w_j} A_j(t)\left(\pi_j^* - r_j(t)\right)$$
$$= \sum_{j=1}^{N} \sqrt{w_j} A_j(t)\left(\frac{\sqrt{w_j}}{\sum\limits_{i=1}^{N} \sqrt{w_i}} - \frac{\alpha^{w_j A_j^2(t)}}{\sum\limits_{i=1}^{N} \alpha^{w_i A_i^2(t)}}\right). \tag{19}$$

Note that we are only interested in the sign of (19), so we can instead look at

$$\sum_{j=1}^{N} \sqrt{w_j} A_j(t)\left(\sqrt{w_j}(\sum_{i=1}^{N} \alpha^{w_i A_i^2(t)}) - \alpha^{w_j A_j^2(t)}(\sum_{i=1}^{N} \sqrt{w_i})\right)$$
$$= \sum_{j=1}^{N} \alpha^{w_j A_j^2(t)}\left(\sum_{i=1}^{N} w_i A_i(t) - \sqrt{w_j} A_j(t) \sum_{i=1}^{N} \sqrt{w_i}\right)$$
$$= \sum_{j=1}^{N} \alpha^{w_j A_j^2(t)}\left(\sum_{i=1}^{N} \sqrt{w_i}\left(\sqrt{w_i} A_i(t) - \sqrt{w_j} A_j(t)\right)\right). \tag{20}$$

Consider without loss of generality that sources are numbered such that $\sqrt{w_1}A_1(t) \ge \sqrt{w_2}A_2(t) \ge \cdots \ge \sqrt{w_N}A_N(t)$. This automatically implies that source 1 has the largest value of $\sqrt{w_j}A_j(t)$ among all sources.

Case 1: First we consider the case when Source 1 is the unique max-weight scheduling decision, i.e. $\pi^{mw}(t) = \underset{j \in [N]}{\operatorname{argmax}}\left(w_j A_j^2(t)\right) = 1$. Since we have assumed weights $w_i \in \mathbb{Z}^+$ and AoIs $A_i(t)$ are also positive integers, the above equation implies that

$$w_1 A_1^2(t) - w_i A_i^2(t) \ge 1, \forall i \ne 1. \tag{21}$$

This is because $w_i A_i^2(t) \in \mathbb{Z}^+$ for all sources i. Further, note that $f(x) = \sqrt{x}$ is Lipschitz for $x \in [1, \infty)$ with the Lipschitz constant 0.5. Applying this fact to (21), we get

$$\sqrt{w_1} A_1(t) - \sqrt{w_i} A_i(t) \ge 0.5, \forall i \ne 1. \tag{22}$$

Next, we define the following quantities

$$\gamma_j \triangleq \left(\sum_{i=1}^{N} \sqrt{w_i}\left(\sqrt{w_i} A_i(t) - \sqrt{w_j} A_j(t)\right)\right). \tag{23}$$

Using (22), it is easy to see that $\gamma_1 \le -0.5\sum_{i=1}^{N} \sqrt{w_i}$. We will bound the rest of the values γ_i in comparison to γ_1.

$$\gamma_j = \left(\sum_{i=1}^{N} \sqrt{w_i}\left(\sqrt{w_i} A_i(t) - \sqrt{w_j} A_j(t)\right)\right)$$
$$\le \left(\sqrt{w_1}(\sqrt{w_1} A_1(t) - \sqrt{w_j} A_j(t))\right.$$
$$+ \sum_{i=2}^{j-1} \sqrt{w_i}\left(\sqrt{w_1} A_1(t) - \sqrt{w_j} A_j(t)\right)$$
$$+ \sum_{i=j}^{N} \sqrt{w_i}\left(\sqrt{w_i} A_i(t) - \sqrt{w_j} A_j(t)\right)\right) \tag{24}$$
$$\le \frac{\sum_{i=1}^{N} \sqrt{w_i}}{\sqrt{w_j}}\left(\sqrt{w_j}(\sqrt{w_1} A_1(t) - \sqrt{w_j} A_j(t))\right.$$
$$+ \sum_{i \ne j} \sqrt{w_i}\left(\sqrt{w_1} A_1(t) - \sqrt{w_i} A_i(t)\right)\right)$$
$$\le \frac{\sum_{i=1}^{N} \sqrt{w_i}}{\min_j \sqrt{w_j}}|\gamma_1|, \forall j \ne 1.$$

Using this result, (21) and the definition of γ_i we get

$$\sum_{j=1}^{N} \alpha^{w_j A_j^2(t)} \left(\sum_{i=1}^{N} \sqrt{w_i} \left(\sqrt{w_i} A_i(t) - \sqrt{w_j} A_j(t) \right) \right)$$

$$\leq \alpha^{w_1 A_1^2(t)} (\gamma_1) + \sum_{i=2}^{N} \alpha^{w_2 A_2^2(t)} \frac{\sum_{i=1}^{N} \sqrt{w_i}}{\min_j \sqrt{w_j}} |\gamma_1| \quad (25)$$

$$\leq \alpha^{w_1 A_1^2(t)} |\gamma_1| \left(-1 + \alpha^{-1} \frac{(N-1) \sum_{i=1}^{N} \sqrt{w_i}}{\min_j \sqrt{w_j}} \right).$$

Clearly, choosing

$$\alpha > \left(\frac{(N-1) \sum_{i=1}^{N} \sqrt{w_i}}{\min_j \sqrt{w_j}} \right), \quad (26)$$

is sufficient to guarantee that (25) is negative and hence (19) is negative. Thus, for this choice of α, we observe that

$$\mathbb{E}\left[\Delta^{CSMA}(t) \Big| \boldsymbol{A(t)} \right] \leq \mathbb{E}\left[\Delta^{SR}(t) \Big| \boldsymbol{A(t)} \right], \forall \boldsymbol{A(t)}. \quad (27)$$

Case 2: Sources $1, \cdots, k$ are all solutions to the following maximization

$$\pi^{mw}(t) = \underset{j \in [N]}{\operatorname{argmax}} \left(w_j A_j^2(t) \right) \in \{1, \cdots, k\}. \quad (28)$$

We can repeat the exact same analysis as Case 1, but starting with

$$w_1 A_1^2(t) - w_i A_i^2(t) \geq 1, \forall i \in \{k+1, \cdots, N\}. \quad (29)$$

This is because $w_i A_i^2(t) \in \mathbb{Z}^+$ for all sources i. Further, note that $f(x) = \sqrt{x}$ is Lipschitz for $x \in [1, \infty)$ with the Lipschitz constant 0.5. Applying this fact to (21), we get

$$\sqrt{w_1} A_1(t) - \sqrt{w_i} A_i(t) \geq 0.5, \forall i \in \{k+1, \cdots, N\}. \quad (30)$$

Using these inequalities above, we obtain

$$\gamma_j \leq \frac{\sum_{i=1}^{N} \sqrt{w_i}}{\min_j \sqrt{w_j}} |\gamma_1|, \forall j \in \{k+1, \cdots, N\}. \quad (31)$$

Finally putting all of the inequalities together, we get

$$\sum_{j=1}^{N} \alpha^{w_j A_j^2(t)} \left(\sum_{i=1}^{N} \sqrt{w_i} \left(\sqrt{w_i} A_i(t) - \sqrt{w_j} A_j(t) \right) \right)$$

$$\leq k \alpha^{w_1 A_1^2(t)} (\gamma_1) + \sum_{i=k+1}^{N} \alpha^{w_{k+1} A_{k+1}^2(t)} \frac{\sum_{i=1}^{N} \sqrt{w_i}}{\min_j \sqrt{w_j}} |\gamma_1|$$

$$\leq \alpha^{w_1 A_1^2(t)} |\gamma_1| \left(-k + \alpha^{-1} \frac{(N-k) \sum_{i=1}^{N} \sqrt{w_i}}{\min_j \sqrt{w_j}} \right). \quad (32)$$

Again, choosing

$$\alpha > \left(\frac{(N-1) \sum_{i=1}^{N} \sqrt{w_i}}{\min_j \sqrt{w_j}} \right) \quad (33)$$

is sufficient to guarantee that (32) is negative and hence (19) is negative. This completes the proof of Lemma 2. $\blacksquare$

Next, we proceed to the proof of Theorem 2. The one-slot drift for the optimal stationary randomized policy is given by -

$$\mathbb{E}\left[\Delta^{sr}(t) \Big| \boldsymbol{A(t)} \right]$$

$$= \sum_{j=1}^{N} \pi_j^* \left(\sqrt{w_j} + \sum_{i \neq j} \sqrt{w_i} (A_i(t) + 1) \right) - \sum_{j=1}^{N} \sqrt{w_j} A_j(t)$$

$$= \sum_{j=1}^{N} \sqrt{w_j} - \sum_{j=1}^{N} \pi_j^* \sqrt{w_j} A_j(t). \quad (34)$$

Putting together (16) and (34), we get

$$\mathbb{E}\left[\Delta^{csma}(t) \Big| \boldsymbol{A(t)} \right] \leq \sum_{j=1}^{N} \sqrt{w_j} - \sum_{j=1}^{N} \pi_j^* \sqrt{w_j} A_j(t). \quad (35)$$

Summing (35) for $t = 1, \cdots, T$ and taking expectation, we get

$$\mathbb{E}\left[L(T+1) - L(1) \right] \leq$$

$$T \sum_{j=1}^{N} \sqrt{w_j} - \sum_{j=1}^{N} \mathbb{E}\left[\sum_{t=1}^{T} \sqrt{w_j} \pi_j^* A_j(t) \right]. \quad (36)$$

Substituting the expression for π_j^* from (4), rearranging and dividing by T, we get

$$\frac{1}{T \left(\sum_{j=1}^{N} \sqrt{w_j} \right)} \mathbb{E}\left[\sum_{t=1}^{T} \sum_{j=1}^{N} w_j A_j(t) \right] \leq$$

$$\sum_{j=1}^{N} \sqrt{w_j} - \frac{1}{T} \mathbb{E}\left[L(T+1) - L(1) \right]. \quad (37)$$

Since $L(T+1) \geq 0$ and $w_j / \pi_j^* = \sqrt{w_j} (\sum_{j=1}^{N} \sqrt{w_j})$, we can further simplify the equaiton above to

$$\frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} \sum_{j=1}^{N} w_j A_j(t) \right] \leq \sum_{j=1}^{N} \frac{w_j}{\pi_j^*} + \frac{\mathbb{E}[L(1)]}{T}. \quad (38)$$

Now, we observe that the average AoI of a source j under the optimal stationary randomized policy is given by $\bar{A}_j^{sr} = 1/\pi_j^*$, as shown in [4]. Using this fact and taking limsup as T goes to infinity in the equation above, we get

$$\sum_{j=1}^{N} w_j \bar{A}_j^{csma}(t) \leq \sum_{j=1}^{N} w_j \bar{A}_j^{sr}. \quad (39)$$

We also know from [4] that stationary randomized policies can be at most a factor of two away from optimal. Thus, we get

$$\frac{\sum_{i=1}^{N} w_i \bar{A}_i^{csma}}{\sum_{i=1}^{N} w_i \bar{A}_i^{opt}} \leq \frac{\sum_{i=1}^{N} w_i \bar{A}_i^{sr}}{\sum_{i=1}^{N} w_i \bar{A}_i^{opt}} \leq 2. \quad (40)$$

This completes the proof. $\blacksquare$

The factor of two optimality guarantee that we have derived is the same performance guarantee as the one shown for the max-weight policy in [29] and better than the factor of four bound derived in [4], [11]. Viewing Theorems 1
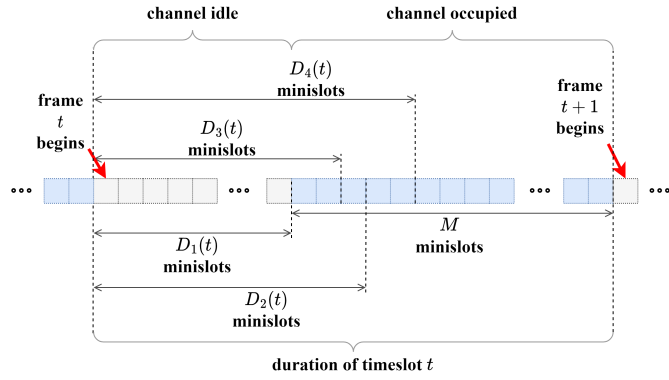
Fig. 2. Events within frame t in the near-realistic multiple access model. Four sources choose backoff timers $D_1(t), \cdots, D_4(t)$. Source 1's timer runs out first, after which it transmits its update for M minislots.

Algorithm 3: Near-realistic Fresh-CSMA

Input : parameters $\alpha > 1$, $\beta > 1$, $B \in \mathbb{Z}^+$

1 **while** $t \in 1, \cdots, T$ **do**
2   **At every source** $i \in 1, \cdots, N$ **do**
3   Generate a random variable

$$Z_i(t) \sim \exp\left(\alpha^{w_i A_i^2(t)}\right).$$

4   Map it to a non-negative integer timer,

$$D_i(t) = \max\left(B + \left\lfloor \log_\beta\left(Z_i(t)\right)\right\rfloor, 0\right).$$

5   **while** $\tau < D_i(t)$ **do**
6   Stay silent
7   **end**
8   **if** *Channel is free* **then**
9   Transmit
10   **end**
11
12 **end**

---

and 2 together, we conclude that the idealized Fresh-CSMA policy can replicate the behavior of the max-weight policy, both at each time-slot with high probability and in terms of long-term average AoI over the entire time-horizon if $\alpha > \max\left((N-1)(1-\delta)/\delta, \frac{(N-1)\sum_{i=1}^{N}\sqrt{w_i}}{\min_j \sqrt{w_j}}\right)$. In Section VI, we will see via simulations that this holds true even for small values of α, i.e., α doesn't need to be very large for Fresh-CSMA to be able to mimic the max-weight policy.

IV. NEAR-REALISTIC MULTIPLE ACCESS MODEL

Until now, we have looked at distributed multiple access with idealized assumptions. In this section, we discuss the Fresh-CSMA protocol under a more realistic version of multiple access.

Our discussion is based on the IEEE 802.11 standard for wireless LAN [30]. This standard defines a distributed coordination function (DCF) for sharing access to the wireless medium based on a CSMA/CA style protocol. The 802.11 standard divides time into the basic units of mini-slots, where each mini-slot is the duration of time needed by a source to detect packet transmission from any another source, i.e., perform channel sensing. A typical value for the mini-slot duration in IEEE 802.11 g/n/ac protocols is $9\mu s$.

Fig. 2 describes the key elements of our near-realistic model. First, we consider that the backoff timers for source i at frame t, denoted by $D_i(t)$, can only run in multiples of minislot durations. This relaxes Assumptions 1 and 2, since the timers are now discrete, with finite precision and limited by the amount of time required to do channel sensing. Further, since timers are no longer continuous, the probability of collision is also non-zero and has an effect on the average age.

As before, we assume that backoff timers for each source begin at the beginning of each frame and the source/sources whose timers run out first get to transmit an entire application layer update. Thus, we have not relaxed Assumption 3 in this model. We assume that a mini-slot takes $1/M$ units of time. So, transmitting an entire update takes M mini-slots. We will discuss in Section VI that for typical values of update sizes and transmission rates M tends to be large (around 10000).

Finally, we also relax Assumption 4 and consider the amount of time that the channel remains idle in each frame. Consider the example frame depicted in Fig. 2 with four backoff timers $D_1(t), \cdots, D_4(t)$. The timer of source 1, denoted by $D_1(t)$, runs out first. Then, source 1 transmits its update for M minislots. Thus the total duration of frame t is $D_1(t) + M$ minislots or alternatively $1 + D_1(t)/M$. However, for the first $D_1(t)$ minislots in frame t, the channel remained idle. We term this the *backoff overhead*. In general, given $D(t) \triangleq \min_{i \in [N]} D_i(t)$, frame t takes $M + D(t)$ minislots or alternatively $1 + D(t)/M$ units of time to complete. Compared to the idealized setting, the time $D(t)/M$ is the backoff overhead of the protocol, since it is the time that the channel must remain idle before any source starts transmitting. We take this overhead into account while calculating AoIs.

Our new model thus relaxes Assumptions 1, 2, and 4, while allowing us to study the effect of collision probabilities and backoff overheads on the AoI. For this near-realistic multiple access model, we provide a modified version of the Fresh-CSMA protocol below. As before, We use t to denote the discrete frames and τ to denote time within each frame, denoting the number of minislots that have passed within this frame.

The key difference between the protocol described in Algorithms 2 and 3 is mapping the continuous random variables $Z_i(t)$ to the discrete variables $D_i(t) \in \{0 \cup \mathbb{Z}^+\}$, which denote the number of mini-slots source i should count down before transmitting.

A. Collisions

Note that when using the protocol above, a packet collision happens if two sources i and j choose the same discrete backoff timers $D_i(t)$ and $D_j(t)$ at frame t while also being the first timers to count down to zero, i.e., $\underset{j \in [N]}{\operatorname{argmin}}\left(D_j(t)\right)$ is not unique. When a collision happens, we assume that the base station fails to receive an update from any of the transmitting

sources and the entire frame is wasted. The following theorem analyzes the probability of the event that two sources i and j choose different backoff timers at time t.

Theorem *3:* Let the AoIs at time t be given by $A_1(t), \cdots, A_N(t)$ and $\lambda_i \triangleq \alpha^{w_i A_i^2(t)}$. Then probability that any two sources i and j choose different backoff timers $D_i(t)$ and $D_j(t)$ can be lower bounded as follows

$$\mathbb{P}\left(D_i(t) \neq D_j(t)\right) \geq \psi\left(B, \beta, \lambda_i, \lambda_j\right) + \psi\left(B, \beta, \lambda_j, \lambda_i\right)$$
(41)

where, the function $\psi(\cdot)$ is given by:

$$\psi\left(B, \beta, \lambda_i, \lambda_j\right) = \frac{\lambda_i e^{-\beta^{-B}(\lambda_i + \beta\lambda_j)}}{\lambda_i + \beta\lambda_j}$$
$$+ \left(e^{\lambda_i \beta^{-B}} - 1\right) e^{-\beta^{-B}(\lambda_i + \beta\lambda_j)}.$$

Proof: Consider the exponential random variables generated by the two sources $Z_i(t) \sim \exp(\lambda_i)$ and $Z_j(t) \sim \exp(\lambda_j)$, where $\lambda_i = \alpha^{w_i A_i^2(t)}, \forall i$. Suppose that the following inequality holds:

$$Z_j(t) > \begin{cases} \beta Z_i(t), & \text{if } Z_i(t) > \beta^{-B} \\ \beta^{-B+1}, & \text{otherwise.} \end{cases}$$
(42)

Then it is easy to see that

$$\max\left(B + \lfloor \log_\beta\left(Z_i(t)\right)\rfloor, 0\right) < \max\left(B + \lfloor \log_\beta\left(Z_j(t)\right)\rfloor, 0\right)$$

which in turn implies that $D_i(t) < D_j(t)$. Switching i and j in the inequality (42), we get $D_i(t) > D_j(t)$. Note that the two events are disjoint.

Thus, we can lower-bound our probability of interest as follows:

$$\mathbb{P}(D_i(t) \neq D_j(t)) \geq \mathbb{P}\left(Z_j(t) > \min\{\beta Z_i(t), \beta^{-B+1}\}\right)$$
$$+ \mathbb{P}\left(Z_i(t) > \min\{\beta Z_j(t), \beta^{-B+1}\}\right).$$
(43)

Simplifying the first term on the RHS, we get

$$\mathbb{P}\left(Z_j(t) > \min\{\beta Z_i(t), \beta^{-B+1}\}\right)$$
$$= \int_0^{\beta^{-B}} \lambda_i e^{-\lambda_i x} e^{-\lambda_j \beta^{-B+1}} dx + \int_{\beta^{-B}}^\infty \lambda_i e^{-\lambda_i x} e^{-\lambda_j \beta x} dx$$
$$= \frac{\lambda_i e^{-\beta^{-B}(\lambda_i + \beta\lambda_j)}}{\lambda_i + \beta\lambda_j} + \left(e^{\lambda_i \beta^{-B}} - 1\right) e^{-\beta^{-B}(\lambda_i + \beta\lambda_j)}$$
$$\triangleq \psi\left(B, \beta, \lambda_i, \lambda_j\right).$$
(44)

By the same argument, the second term on the RHS of (43) is equal to $\psi\left(B, \beta, \lambda_j, \lambda_i\right)$. Thus, we get

$$\mathbb{P}(D_i(t) \neq D_j(t)) \geq \psi\left(B, \beta, \lambda_i, \lambda_j\right) + \psi\left(B, \beta, \lambda_j, \lambda_i\right). \quad (45)$$

This completes the proof.   ∎

As a corollary of this proof, note that

$$\frac{\partial}{\partial B} \psi\left(B, \beta, \lambda_i, \lambda_j\right)$$
$$= \lambda_j \beta^{-B+1} \log(\beta)\left(e^{\lambda_i \beta^{-B}} - 1\right) e^{-\beta^{-B}(\lambda_i + \beta\lambda_j)} \quad (46)$$
$$\geq 0.$$

The last inequality follows since $\beta > 1$ and $\lambda_i > 0$. Thus, the probability that two sources choose different backoff timers *increases with the parameter B*. In the limit as $B \to \infty$, we get

$$\lim_{B \to \infty} \mathbb{P}(D_i(t) \neq D_j(t)) \geq \frac{\lambda_i}{\lambda_i + \beta\lambda_j} + \frac{\lambda_j}{\lambda_j + \beta\lambda_i}. \quad (47)$$

Thus, when B is large, the probability that two sources occupy different mini-slots *decreases with β*. Putting the two observations together, we should choose a large value for B and a small value for β (close to 1) to reduce collisions. However, for finite B, (47) does not hold and the value of β cannot be too small, since in that case, all the discrete timers will map to the first minislot leading to collisions in almost every frame.

In the analysis above, we used the probability of two sources occupying different backoff timers as a proxy for analyzing the collision probability directly, since a tight bound for the actual collision probability is too involved to compute. In Section VI, we will see via simulations how the actual collision probability varies with the parameters B and β.

B. Backoff Timer Overhead

Next, we analyze the overhead of the backoff timers in the Fresh-CSMA protocol in the near-realistic model. This is unlike the idealized setting where we ignored the time taken by the backoff timers to count down, during which the channel remains idle.

Recall that the quantity $D(t)/M$ is what we defined as the backoff overhead of a protocol at time t, since it is the time that the channel must remain idle before any source starts transmitting. The following theorem provides an upper-bound on the expected backoff overhead in frame t.

Theorem *4:* Let the AoIs at time t be $A_1(t), \cdots, A_N(t)$ and $\lambda_i \triangleq \alpha^{w_i A_i^2(t)}$. Then, the expected idle-time of the Fresh-CSMA protocol at time t can be upper-bounded by

$$\frac{1}{M} \mathbb{E}\left[D(t)\right] \leq \frac{1}{M} + \frac{\Gamma\left(0, \lambda\beta^{-B}\right)}{M \log(\beta)}. \quad (48)$$

Here $\Gamma(\cdot, \cdot)$ is the upper incomplete gamma function and $\lambda \triangleq \sum_{i \in [N]} \lambda_i$.

Proof: Let $Z(t) = \min_{i \in [N]} Z_i(t)$. Since $Z(t)$ is the minimum of N independent exponential random variables, it is also

exponentially distributed with the parameter $\lambda = \sum_{i\in[N]}\lambda_i$. Using this, we provide an upper bound for $\mathbb{E}[D(t)]$ below.

$$
\begin{aligned}
\mathbb{E}\big[D(t)\big] &= \mathbb{E}\bigg[\min_{i\in[N]}\big(D_i(t)\big)\bigg]\\
&= \mathbb{E}\bigg[\min_{i\in[N]}\Big(\max\big(B+\lfloor\log_\beta\big(Z_i(t)\big)\rfloor,0\big)\Big)\bigg]\\
&\le B+1+\mathbb{E}\bigg[\max\Big(\log_\beta\big(\min_{i\in[N]}Z_i(t)\big),-B\Big)\bigg]\\
&\le B+1+\mathbb{E}\bigg[\max\Big(\log_\beta\big(Z(t)\big),-B\Big)\bigg]\\
&\le 1+\frac{\Gamma\big(0,\lambda\beta^{-B}\big)}{\log(\beta)}
\end{aligned}
$$

(49)

The last inequality follows from the fact that $\mathbb{E}\big[\max(\log(Z(t)),-B)\big] = -B + \Gamma\big(0,\lambda\beta^{-B}\big)$, where $\Gamma(\cdot,\cdot)$ is the upper incomplete gamma function given by

$$
\Gamma(s,x) = \int_x^\infty t^{s-1}e^{-t}dt. \tag{50}
$$

∎

Note that the function $\Gamma(0,x)$ is given by

$$
\Gamma(0,x) = \int_x^\infty t^{-1}e^{-t}dt. \tag{51}
$$

From (51), note that $\Gamma(0,x)$ is decreasing in x. Thus, for a fixed value of β, the expected idle time *increases as B increases*. The exact dependence on β is more tricky to evaluate, so we again consider the case of large B as an alternative. First, we make the following observation for the gamma function of large values of B

$$
\Gamma(0,\lambda\beta^{-B}) \approx B\log(\beta) - log(\lambda) - \gamma, \tag{52}
$$

where $\gamma \approx 0.58$ is the Euler-Mascheroni constant. Using this, it is easy to see that for large values of B, the expected idle time upper bound is approximately equal to $1+B-\log_\beta(\lambda)$. Thus, the backoff overhead also *increases with β*, given a fixed large value of B.

Together this implies that we need to choose a relatively small value of B and a small value of β (close to 1) to reduce idle time. Importantly, there is a tradeoff between the collision probability and the backoff overhead depending on the choice of the parameter B. A larger value of B reduces the probability of collision but at the cost of higher backoff overhead.

Theorem 4 also allows us to compute an approximate upper-bound for the average idle time over the entire horizon. Suppose the average AoI of source i under the Fresh-CSMA policy in the near-realistic model is denoted by $\bar{A}_i$. Using the average AoIs, we define the following quantity: $\bar{\lambda} \triangleq \sum_{i=1}^N \alpha^{w_i\bar{A}_i^2}$. Then, an approximate upper-bound of the average backoff overhead per frame over the entire time-horizon $\bar{D}^{ub}$ can be obtained as follows:

$$
\bar{D}^{ub} \approx \frac{1}{M} + \frac{\Gamma\big(0,\bar{\lambda}\beta^{-B}\big)}{M\log(\beta)}. \tag{53}
$$

In Section VI, we will see via simulations that this is a good bound for the average backoff overhead in the near-realistic setting.

V. GOING BEYOND AGE OF INFORMATION

While AoI has been used as a proxy for optimizing monitoring and control costs in real-time settings over the past decade, a recent line of work starting with [31] has proposed a new and more general metric to measure the impact of stale information on underlying real-time monitoring tasks. This metric is called the AoII, and it has been used to study the monitoring of Markov sources in various kinds of settings [31]–[34].

The key idea behind the AoII metric is that takes into account the actual error or distortion between the estimate of the process being monitored at the remote monitor and the actual value of the process at present. More precisely, suppose that $X(t)$ represents the state of the process that needs to be monitored and let $\hat{X}(t)$ be the estimate of the process at time t at the remote monitor. Let the function $g(X(t),\hat{X}(t))$ measure the distortion or error between the actual process and its remote estimate and let $f(\cdot)$ be a monotone increasing function. Let $V(t)$ represent the most recent time instant up to the current time t at which this distortion was zero. The AoII is then defined as

$$
AoII(t) \triangleq f\big(t-V(t)\big)g(X(t),\hat{X}(t)). \tag{54}
$$

Note that if the actual process and its estimate stay the same for some time despite no new updates being delivered to the monitor, the AoII remains zero while the AoI keeps increasing. Thus, the AoII can be viewed as a more accurate metric to measure information uncertainty at the monitor.

Now, consider a setting with N sources, sending updates to the base station, where only one source can talk to the base station at any given time. Unlike the setting we have analyzed until now, we will now look at minimizing the sum of AoIIs instead of weighted AoIs. Each source is tracking a process $X_i(t), i \in \{1,\cdots,N\}$ and the base station maintains estimates for each process $\hat{X}_i(t), i \in \{1,\cdots,N\}$. Using these estimates and 54, we can compute the AoIIs for each source, denoted by $AoII_i(t), i \in \{1,\cdots,N\}$. We want to design a scheduling policy that minimizes the long term time-average of the AoIIs, i.e.

$$
\operatorname*{argmin}_{\pi}\bigg(\limsup_{T\to\infty}\bigg[\frac{1}{T}\frac{1}{N}\sum_{t=1}^T\sum_{i=1}^N AoII_i(t)\bigg]\bigg). \tag{55}
$$

Optimal scheduling design in this setting would crucially need to take the dynamics of the underlying sources into account. This is especially true in asymmetric cases, when some sources evolve much faster than others. In settings where source dynamics are fairly similar, a good heuristic candidate policy to solve this problem would be to schedule the source with the highest AoII at each time-slot.

$$
\pi^{max-AoII}(t) = \operatorname*{argmax}_{j\in[N]}\bigg(AoII_j(t)\bigg). \tag{56}
$$

However, a crucial drawback of the AoII metric is the fact that its computation requires knowledge of the actual current state of the process $X(t)$. Thus, AoIIs cannot be computed at the base station beforehand and a centralized multiple source scheduling policy like (56) cannot be implemented in reality, since the base station does not know the actual current states of each process. The CSMA based protocols we develop in this

---

Algorithm 4: Idealized Fresh-CSMA with AoIIs

Input : parameters $\alpha > 1, \delta \ll 1$

1 while $t \in 1, \cdots, T$ **do**

2   At every source $i \in 1, \cdots, N$ **do**

3   Generate a random timer
$$Z_i(t) \sim \exp\left(\alpha^{AoII_i(t)}\right).$$

4   **while** $\tau < \delta Z_i(t)$ **do**

5   Stay silent

6   **end**

7   **if** *Channel is free* **then**

8   Transmit

9   **end**

10

11 end

---

work provide a way out of this dilemma. Sources can compute their own AoIIs, since they have access to $X(t)$, $\hat{X}(t)$, t, and $V(t)$. Then, a CSMA style policy that uses AoIIs instead of the AoIs can be implemented to pick the source that has the highest AoII and get better monitoring performance. We illustrate how to incorporate AoII into a CSMA style policy using the idealized version of Fresh-CSMA in Algorithm 4. The near-realistic version follows immediately, by replacing the weighted AoI with the AoII.

For the setting involving monitoring Markov sources, the following choice of the functions $f(\cdot)$ and $g(\cdot)$ is typically considered in literature

$$AoII(t) = \big(t - V(t)\big)\mathbb{1}_{\{X(t) \neq \hat{X}(t)\}}. \tag{57}$$

For this specific AoII metric, we can show a result similar to Theorem 1 in the case of weighted AoI.

Theorem 5: Given any $\delta \in (0, 1)$ and $AoII_1(t), \cdots, AoII_N(t)$ evolving according to 57; if we set $\alpha \geq (N-1)\frac{1-\delta}{\delta}$, then the following holds

$$\mathbb{P}\left(\pi^{CSMA-AoII}(t) = \pi^{max-AoII}(t)\right) \geq 1 - \delta. \tag{58}$$

Here, $\pi^{max-AoII}(t)$ is the scheduling decision given by (56), while $\pi^{CSMA-AoII}(t)$ is the scheduling decision made by the idealized Fresh-CSMA policy that utilizes AoIIs (Algorithm 4)

Proof: The proof is identical to that of Theorem 1 - all we need to do is replace the weighted AoIs with the AoIIs. The evolution of AoII according to 57 ensures that all the AoII values are integers. As before, we divide the proof into two parts.

Case 1: The expression $\underset{j \in [N]}{\operatorname{argmax}}\Big(AoII_j(t)\Big)$ has a unique maximum. Let this maximum be the source 1 without loss of generality. Then, the max-AoII decision is to schedule source 1. Since we know that AoIIs are integers by (57), so the

following must hold:

$$AoII_1(t) - AoII_i(t) \geq 1, \forall i \neq 1. \tag{59}$$

Now, applying Lemma 1, we can calculate the following probability

$$\mathbb{P}\left(\pi^{CSMA-AoII}(t) = 1\right) = \frac{\alpha^{AoII_1(t)}}{\sum_{j=1}^{N} \alpha^{AoII_j(t)}}$$

$$= \frac{1}{1 + \sum_{i=2}^{N} \alpha^{AoII_i(t)-AoII_1(t)}}$$

$$\geq \frac{1}{1 + (N-1)\alpha^{-1}}$$

$$\geq 1 - \delta. \tag{60}$$

The first inequality follows by using (59) while the second inequality follows due to the fact that $\alpha \geq (N-1)(1-\delta)/\delta$.

Case 2: The expression $\underset{j \in [N]}{\operatorname{argmax}}\Big(AoII_j(t)\Big)$ has multiple maxima. Suppose that the set of maxima is given by the nodes $\{1, \cdots, k\}$. Then, the max-AoII policy will choose one of these sources to be scheduled. We want to lower bound the probability that Fresh-CSMA with AoIIs chooses a node from within this set. To do so, we first make a similar observation as in the case above.

$$AoII_1(t) - AoII_j(t) \geq 1, \forall j = k+1, \cdots, N. \tag{61}$$

Using Lemma 1, we calculate the probability of interest

$$\mathbb{P}\left(\pi^{CSMA-AoII}(t) \in \{1, \cdots, k\}\right) = \frac{k\alpha^{AoII_1(t)}}{\sum_{j=1}^{N} \alpha^{AoII_j(t)}}$$

$$= \frac{k}{k + \sum_{i=k+1}^{N} \alpha^{AoII_i(t)-AoII_1(t)}} \tag{62}$$

$$\geq \frac{1}{1 + (N-k)\alpha^{-1}}$$

$$\geq 1 - \delta.$$

As before, the first inequality follows by using (61) while the second inequality follows due to the fact that $\alpha \geq (N-1)(1-\delta)/\delta$. This completes the proof. ∎

Theorem 5 shows that on a per-time-slot basis, Fresh-CSMA based on AoII matches the scheduling decisions made by the hypothetical centralized max-AoII policy (56), which cannot be implemented in reality. In Section VI, we will show via simulations that the Fresh-CSMA policy based on AoII can achieve lower time-average AoII across the network than even the centralized max-weight policy that utilizes only AoIs. This suggests that in certain settings distributed policies that utilize monitoring error can outperform centralized policies that have to make scheduling decisions while being oblivious to the actual processes.

VI. NUMERICAL RESULTS

To verify the theoretical results developed in earlier sections, we now provide numerical results from packet level
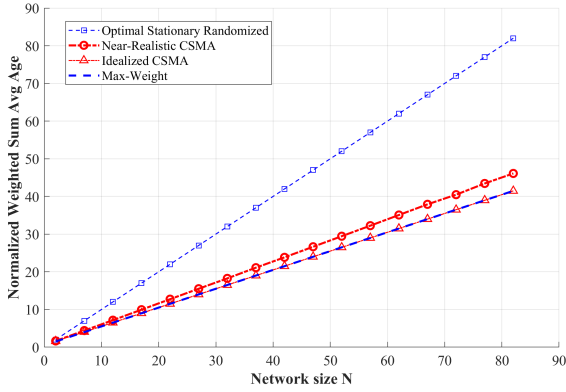
Fig. 3. Normalized average AoI vs. system size (N) for symmetric weights.
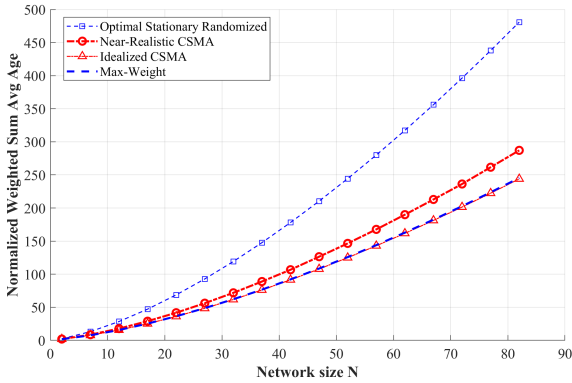

Fig. 4. Normalized average AoI vs. system size (N) for unequal weights.

simulations of all the policies. Throughout this section, we assume that the minislot is $9\mu s$ long (a typical value in IEEE 802.11 implementations [30]) and an update packet from a each source is roughly 600 kB.

Thus, at a data rate of 54 Mbps (which is the highest data rate for IEEE 802.11g in the 2.4 GHz band [30]), it takes 10000 minislots to finish sending an update. This, in turn, implies that we set $M = 10000$ for our simulations, i.e., each packet takes 10000 minislots to transmit.

Each experiment involves calculating the time-average of AoI, AoII, collision probabilities or backoff timer overheads. All of these time-averages are reported for experiments that involve 100000 application layer update packets being transmitted. Unless otherwise specified, we set $\alpha = 1 + \frac{1}{\sum_i w_i}$, $\beta = 1.1 + \max(\log(\log(N)), 0)$, and $B = 250 + N$ for the near-realistic CSMA implementation.

First, we consider the weighted sum average age minimization problem in the setting with equal weights, i.e., $w_i = 1, \forall i$. Fig. 3 plots the performance of two centralized policies - the optimal stationary randomized policy and the max-weight policy, as well as two distributed policies - the idealized Fresh-CSMA protocol and the near-realistic Fresh-CSMA protocol, as the the number of sources in the system N increases. We plot the normalized weighted sum average age, i.e., $(1/N) \sum_i w_i \bar{A}_i$ for each policy. We observe that the idealized Fresh-CSMA protocol matches the performance of the max-weight policy almost exactly, as expected from Theorems 1 and 2. Further, the near-realistic Fresh-CSMA
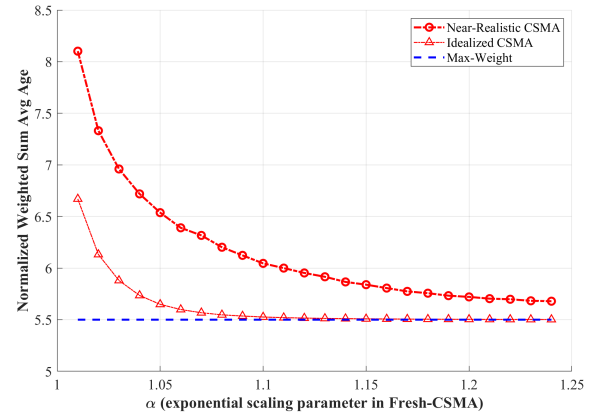

Fig. 5. Normalized average AoI vs. α.

protocol has only a small performance gap to the max-weight policy (due to collisions and backoff overheads) but still significantly outperforms the optimal stationary randomized policy. Note that we use an idealized implementation of the centralized max-weight policy that is assumed to have no polling/signaling overhead. A more realistic implementation would only make the performance gap smaller.

Next, we consider the same setting but with asymmetric weights. We set the weight for source k to be $\sqrt{k}$, i.e., $w_k = \sqrt{k}$, and plot the normalized average age as the system size N increases in Fig. 4. We make the same observations regarding the performance of the policies as in the case of symmetric weights and further note that while our theoretical results needed weights w_i to be integers, that assumption is not required to get good performance in practice.

Next, we consider how the performance of our proposed protocols depends on the parameter α. Theorems 1 and 2 suggest that α needs to be very large to guarantee performance of Fresh-CSMA close to max-weight. However, we find that Fresh-CSMA starts performing very similarly to max-weight even at very small values of α, as show in Fig. 5 for a symmetric system with $N = 10$ sources. This is important in practice since large values of α could lead to integer overflows. As expected, the performance gap narrows as α increases.

Next, we look at the near-realistic Fresh-CSMA protocol in detail. In Fig. 6, we plot the collision probability for this protocol in a symmetric system with $N = 10$ sources as β is varied, while fixing all other parameters. We observe that for $\beta < 1.05$, the collision probability is 1 since all timers map to the first mini-slot. However, as we increase β beyond 1.05, we observe that the collision probability first drops to 0.01 and then gradually increases to 0.06. In Fig. 7, we plot the collision probability as B is varied, while fixing all other parameters. For small values of B, the collision probability is almost 1, but as B increases beyond a threshold, the collision probability stays roughly constant at around 0.015. These results are in line with what we expected from Theorem 3.

Next, we look at the average overhead of near-realistic Fresh-CSMA. Figs. 8 and 9 plot the average overhead (in number of minislots) as the parameters β and B are varied, respectively. As expected from Theorem 4, the overhead increases with both β and B. We note that our approximate
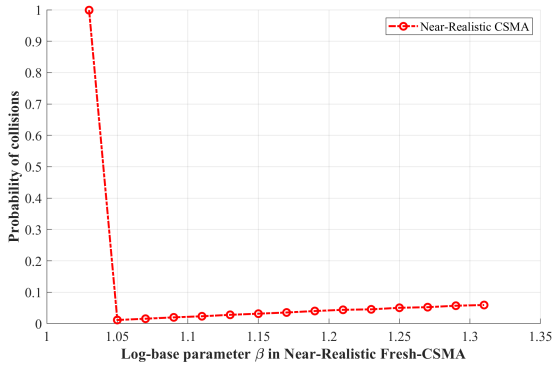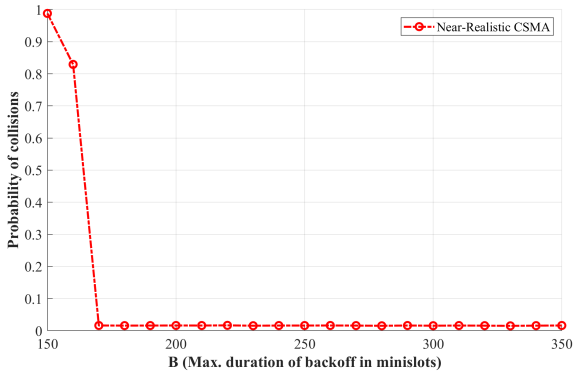
Fig. 6. Probability of collisions vs. β.

Fig. 7. Probability of collisions vs. B.

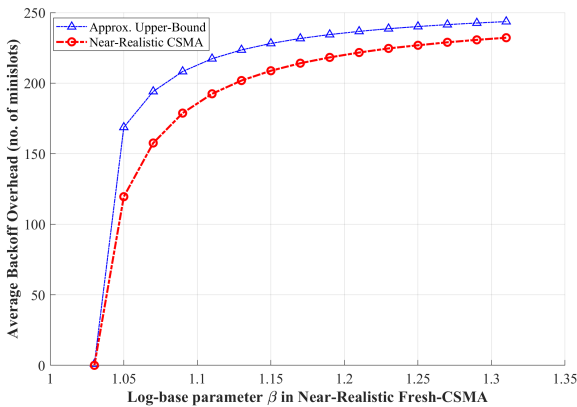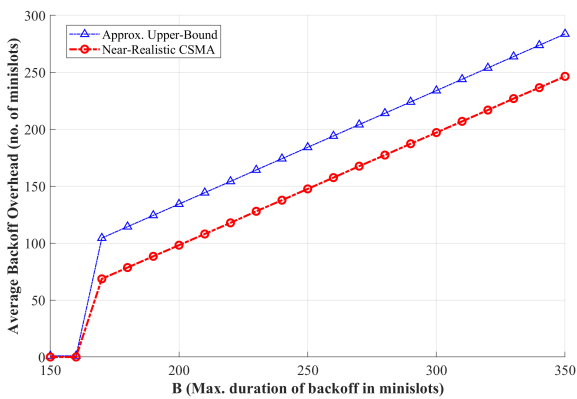Fig. 8. Average backoff overhead vs. β.
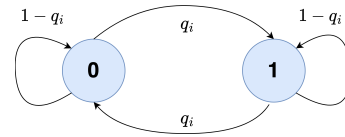

Fig. 9. Average backoff overhead vs. B.

Fig. 10. Symmetric two-state Markov chain representing the ith source.
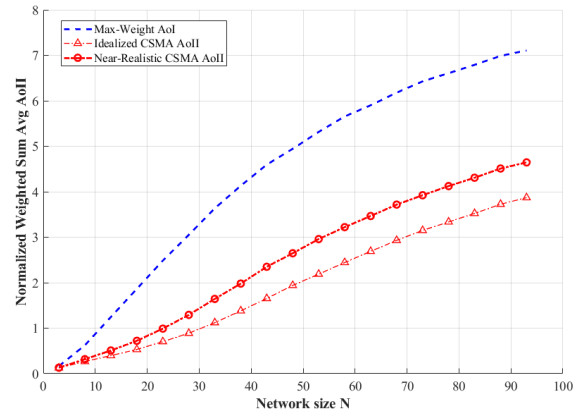

Fig. 11. Normalized average AoII vs system size N when symmetric Markov sources.
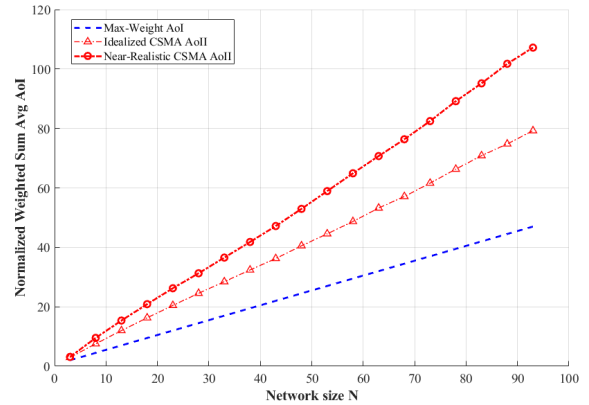

Fig. 12. Normalized average AoI vs system size N when monitoring symmetric Markov sources.

expression for the overhead (53) is a good upper-bound that can be used in practice for system design. Also, we observe that the overhead remains relatively small (2–3%) compared to the update size, which takes 10000 minislots.

Finally, we consider the AoII metric discussed in Section V. In this context, we look at the setting where each source is a symmetric two-state Markov chain, evolving independently over time (see Fig. 10). We set the transition probability q_i for each source to be 0.05 and implement the following three policies - centralized max-weight that uses AoI, distributed idealized Fresh-CSMA that uses AoII and distributed near-realistic Fresh-CSMA that uses AoII. For the CSMA implementations, we set the parameters as follows $\alpha = 2.1$, $\beta = 1.05 + \log(\log(N))$ and $B = 250 + \lfloor N/4 \rfloor$.

Fig. 11 plots the time-average AoII performance of the three policies as we increase the number of sources in the system N. Here AoII is computed using (57). Clearly, the distributed CSMA versions deliver much better AoII performance than the centralized policy that is only able to utilize AoIs. Specifically,

for 93 sources, the idealized Fresh-CSMA with AoII performs about 45% better than AoI max-weight and the near-realistic version of Fresh-CSMA with AoII performs about 35% better than AoI max-weight. The main reason behind this performance improvement is the fact that the distributed policy takes AoII into account for scheduling while the centralized policy can only take AoI into account. This suggests that our CSMA based design is general and can easily accommodate other kinds of information freshness and distortion metrics.

Interestingly, the gain in AoII performance comes at the cost of higher AoIs. In Fig. 12, we plot the long-term time-average AoIs for the same three policies while monitoring Markov sources as the number of source N is increased. We observe that the distributed CSMA versions have higher AoIs than the centralized AoI max-weight. So, Fresh-CSMA based on AoII trades off better monitoring performance/error measured in terms of AoII with worse performance in terms of standard AoI.

VII. CONCLUSION

In this work, we designed a distributed CSMA protocol to minimize weighted sum Age of Information in single-hop wireless networks. We showed that under idealized assumptions, our proposed protocol can closely replicate the behavior of centralized policies known to be nearly optimal. We also analyzed our protocol under a near-realistic medium access model and showed how system parameter choices affect packet collisions and overhead. Our simulation results confirm that our protocol works well in practice and that the performance gap between the idealized version and the near-realistic version of Fresh-CSMA is small. We have also extended some of our results to AoII, a more general information freshness metric. Two important directions of work involve further analysis of our protocols beyond AoI and AoII to metrics such as monitoring error or real-time control costs, and implementing the protocol in real systems to compare performance against standard WiFi.

REFERENCES

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

[2] C. Kam, S. Kompella, and A. Ephremides, "Age of information under random updates," in *Proc. IEEE ISIT*, 2013.

[3] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Trans. Inf. Theory*, vol. 63, pp. 7492–7508, Nov. 2017.

[4] 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, 2018.

[5] I. Kadota, A. Sinha, and E. Modiano, "Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1359–1372, 2019.

[6] R. Talak, S. Karaman, and E. Modiano, "Optimizing information freshness in wireless networks under general interference constraints," in *Proc. ACM MobiHoc*, 2018.

[7] A. Maatouk, S. Kriouile, M. Assad, and A. Ephremides, "On the optimality of the whittle's index policy for minimizing age of information," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1263–1277, 2020.

[8] C. Li, S. Li, and Y. T. Hou, "A general model for minimizing age of information at network edge," in *Proc. IEEE INFOCOM*, 2019.

[9] P. R. Jhunjhunwala and S. Moharir, "Age-of-information aware scheduling," in *Proc. IEEE SPCOM*, 2018.

[10] V. Tripathi and E. Modiano, "A whittle index approach to minimizing functions of age of information," in *Proc. IEEE Allerton*, 2019.

[11] R. Talak, S. Karaman, and E. Modiano, "Distributed scheduling algorithms for optimizing information freshness in wireless networks," in *Proc. IEEE SPAWC*, 2018.

[12] Z. Jiang, B. Krishnamachari, S. Zhou, and Z. Niu, "Can decentralized status update achieve universally near-optimal age-of-information in wireless multiaccess channels?," in *Proc. IEEE ITC*, 2018.

[13] X. Chen, K. Gatsis, H. Hassani, and S. S. Bidokhti, "Age of information in random access channels," *IEEE Trans. Inf. Theory*, 2022.

[14] X. Chen, X. Liao, and S. S. Bidokhti, "Real-time sampling and estimation on random access channels: Age of information and beyond," in *Proc. IEEE INFOCOM*, 2021.

[15] O. T. Yavascan and E. Uysal, "Analysis of age-aware slotted aloha," in *Proc. IEEE Globecom Workshop*, 2020.

[16] 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, 2021.

[17] A. Maatouk, M. Assaad, and A. Ephremides, "On the age of information in a CSMA environment," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 818–831, 2020.

[18] A. Baiocchi and I. Turcanu, "Age of information of one-hop broadcast communications in a CSMA network," *IEEE Commun. Lett.*, vol. 25, no. 1, pp. 294–298, 2020.

[19] M. Wang and Y. Dong, "Broadcast age of information in CSMA/CA based wireless networks," in *Proc. IEEE IWCMC*, 2019.

[20] A. M. Bedewy, Y. Sun, R. Singh, and N. B. Shroff, "Optimizing information freshness using low-power status updates via sleep-wake scheduling," in *Proc. ACM MobiHoc*, 2020.

[21] A. M. Bedewy, Y. Sun, R. Singh, and N. B. Shroff, "Low-power status updates via sleep-wake scheduling," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 2129–2141, 2021.

[22] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part i-carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. 23, no. 12, pp. 1400–1416, 1975.

[23] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.

[24] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 960–972, 2009.

[25] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling," *IEEE Trans. Inf. Theory*, vol. 58, no. 10, pp. 6541–6555, 2012.

[26] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 825–836, 2011.

[27] B. Li and A. Eryilmaz, "Optimal distributed scheduling under time-varying conditions: A fast-CSMA algorithm with applications," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3278–3288, 2013.

[28] V. Tripathi, N. Jones, and E. Modiano, "Fresh-CSMA: A distributed protocol for minimizing age of information," in *Proc. IEEE INFOCOM*, 2023.

[29] Y. Sun, I. Kadota, R. Talak, and E. Modiano, "Age of information: A new metric for information freshness," *Synthesis Lectures on Communication Networks*, vol. 12, no. 2, pp. 1–224, 2019.

[30] "IEEE standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp. 1–4379, 2020.

[31] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The age of incorrect information: A new performance metric for status updates," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2215–2228, 2020.

[32] C. Kam, S. Kompella, and A. Ephremides, "Age of incorrect information for remote estimation of a binary Markov source," in *Proc. IEEE INFOCOM Workshop*, 2020.

[33] S. Kriouile and M. Assaad, "Minimizing the age of incorrect information for real-time tracking of markov remote sources," in *Proc. IEEE ISIT*, 2021.

[34] A. Maatouk, M. Assaad, and A. Ephremides, "The age of incorrect information: An enabler of semantics-empowered communication," *IEEE Trans. Wireless Commun.*, 2022.

Vishrant Tripathi is a PhD student at the Laboratory of Information and Decision Systems (LIDS), at the Massachusetts Institute of Technology (MIT). Prior to this, he received a Bachelors in Technology (B.Tech.) degree from the department of Electrical Engineering at Indian Institute of Technology, Bombay (IIT-B), India, in 2017.

His research focuses on the 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. He was a Recipient of the ACM MobiHoc 2022 Best Paper Runner-Up Award.

Nicholas Jones received the B.S. degree in Electrical Engineering, magna cum laude, from the University of Notre Dame in 2019, and the S.M. degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (MIT) in 2022. He is currently pursuing a Ph.D. in the Laboratory for Information and Decision Systems (LIDS) at MIT. His research involves modeling, analysis, and design of communication networks, with a particular interest in optimizing performance of wireless networks for real-time applications.

Eytan Modiano is The Richard C. Maclaurin Professor in the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems (LIDS) at MIT. Prior to Joining the Faculty at MIT in 1999, he was a Naval Research Laboratory Fellow between 1987 and 1992, a National Research Council Post Doctoral Fellow during 1992-1993, and a Member of the technical staff at MIT Lincoln Laboratory between 1993 and 1999. Eytan Modiano received his B.S. degree in Electrical Engineering and Computer Science from the University of Connecticut at Storrs in 1986 and his M.S. and PhD degrees, both in Electrical Engineering, from the University of Maryland, College Park, MD, in 1989 and 1992, respectively.

His research is on modeling, analysis and design of communication networks and protocols. He received the Infocom Achievement Award (2020) for contributions to the analysis and design of cross-layer resource allocation algorithms for wireless, optical, and satellite networks. He is the 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 Editor-in-Chief for IEEE/ACM Transactions on Networking (2017-2020), and served as Associate Editor for IEEE Transactions on Information Theory and IEEE/ACM Transactions on Networking. He was the Technical Program Co-chair for IEEE Wiopt 2006, IEEE Infocom 2007, ACM MobiHoc 2007, and DRCN 2015; and general Co-chair of Wiopt 2021. He had served on the IEEE Fellows committee in 2014 and 2015, and is a Fellow of the IEEE and an Associate Fellow of the AIAA.