Scheduling Policies for Minimizing Age of Information in Broadcast Wireless Networks

Igor Kadota, Abhishek Sinha, Elif Uysal-Biyikoglu, Rahul Singh, and Eytan Modiano, *Fellow, IEEE*

***Abstract*— In this paper, we consider a wireless broadcast network with a base station sending time-sensitive information to a number of clients through unreliable channels. The Age of Information (AoI), namely the amount of time that elapsed since the most recently delivered packet was generated, captures the freshness of the information. We formulate a discrete-time decision problem to find a transmission scheduling policy that minimizes the expected weighted sum AoI of the clients in the network. We first show that in symmetric networks, a greedy policy, which transmits the packet for the client with the highest current age, is optimal. For general networks, we develop three low-complexity scheduling policies: a randomized policy, a Max-Weight policy and a Whittle's Index policy, and derive performance guarantees as a function of the network configuration. To the best of our knowledge, this is the first work to derive performance guarantees for scheduling policies that attempt to minimize AoI in wireless networks with unreliable channels. Numerical results show that both the Max-Weight and Whittle's Index policies outperform the other scheduling policies in every configuration simulated, and achieve near optimal performance.**

***Index Terms*— Age of Information, scheduling, optimization, quality of service, wireless networks.**

I. INTRODUCTION

AGE OF Information (AoI) has been receiving increasing attention in the literature [2]–[25], particularly for applications that generate time-sensitive information such as position, command and control, or sensor data. An interesting feature of this performance metric is that it captures the freshness of the information *from the perspective of the destination*, in contrast to the long-established packet delay, that represents the freshness of the information with respect to individual packets. In particular, AoI measures the *time that elapsed since the generation of the packet that was most recently delivered to the destination*, while packet delay measures the time interval between the generation of a packet and its delivery.

TABLE I

EXPECTED DELAY, EXPECTED INTER-DELIVERY TIME AND AVERAGE AOI OF A M/M/1 QUEUE WITH $\mu = 1$ AND VARIABLE λ

λ (pkt/sec)	$\mathbb{E}$[delay] (sec)	$\mathbb{E}$[inter-delivery] (sec)	Average AoI (sec)
0.01	1.01	**100.00**	**101.00**
0.53	2.13	1.89	3.48
0.99	**100.00**	1.01	**100.02**

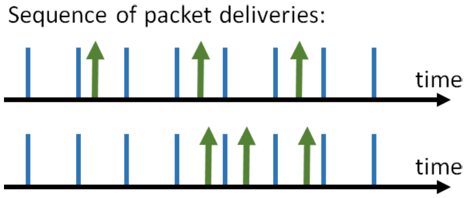

Fig. 1. Two sample sequences of packet deliveries are represented by the green arrows. Both sequences have the same throughput, namely 3 packets over the interval, but different delivery regularity.

The two parameters that influence AoI are packet delay and packet inter-delivery time. In general, controlling only one is insufficient for achieving good AoI performance. For example, consider an M/M/1 queue with a low arrival rate and a high service rate. In this setting, the queue is often empty, resulting in low packet delay. Nonetheless, the AoI can still be high, since infrequent packet arrivals result in outdated information at the destination. Table I provides a numerical example of an M/M/1 queue with fixed service rate $\mu = 1$ and a variable arrival rate λ. The first and third rows represent a system with a high average AoI caused by high inter-delivery time and high packet delay, respectively. The second row shows the queue at the point of minimum average AoI [2].

A good AoI performance is achieved when packets with low delay are delivered regularly. It is important to emphasize the difference between delivering packets regularly and providing a minimum throughput. Figure 1 illustrates the case of two sequences of packet deliveries that have the same throughput but different delivery regularity. In general, a minimum throughput requirement can be fulfilled even if long periods with no delivery occur, as long as those are balanced by periods of consecutive deliveries.

The problem of minimizing AoI was introduced in [2] and has been explored using different approaches. Queueing Theory is used in [2]–[9] for finding the optimal server utilization with respect to AoI. The authors in [10]–[13] consider the problem of optimizing the times in which packets are generated at the source in networks with energy-harvesting or maximum update frequency constraints. Link scheduling optimization with respect to AoI has been

Manuscript received October 27, 2017; revised July 19, 2018; accepted September 18, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. Eryilmaz. Date of publication October 30, 2018; date of current version December 14, 2018. This work was supported in part by NSF under Grant AST-1547331, Grant CNS-1713725, and Grant CNS-1701964; in part by the Army Research Office (ARO) under Grant W911NF-17-1-0508; in part by METU; and in part by the CAPES/Brazil. This paper was presented in part at the Allerton Conference in 2016. *(Corresponding author: Igor Kadota.)*

I. Kadota, A. Sinha, R. Singh, and E. Modiano are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA, and also with Middle East Technical University, 06800 Ankara, Turkey (e-mail: kadota@mit.edu; sinhaa@mit.edu; rsingh12@mit.edu; modiano@mit.edu).

E. Uysal-Biyikoglu is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA, and also with the Department of Electrical Engineering, Middle East Technical University, 06800 Ankara, Turkey (e-mail: uelif@metu.edu.tr).

Digital Object Identifier 10.1109/TNET.2018.2873606

recently considered in [14]–[21]. Applications of AoI are studied in [22]–[25].

The problem of optimizing link scheduling decisions in broadcast wireless networks with respect to throughput and delivery times has been studied extensively in the literature. Throughput maximization of traffic with strict packet delay constraints has been addressed in [26]–[29]. Inter-delivery time is considered in [30]–[36] as a measure of service regularity. Age of Information has been considered in [14]–[21].

In this paper, we consider a network in which packets are generated periodically and transmitted through unreliable channels. Minimizing the AoI is particularly challenging in wireless networks with unreliable channels due to transmission errors that result in packet losses. Our main contribution is the development and analysis of four low-complexity scheduling policies: a Greedy policy, a randomized policy, a Max-Weight policy and a Whittle's Index policy. We first show that Greedy achieves minimum AoI in symmetric networks. Then, for general networks, we compare the performance of each policy against the optimal AoI and derive the corresponding performance guarantees. To the best of our knowledge, this is the first work to derive performance guarantees for policies that attempt to minimize AoI in wireless networks with unreliable channels. A preliminary version of this work appeared in [1].

The remainder of this paper is outlined as follows. In Sec. II, the network model is presented. In Sec. III, we find the optimal scheduling policy for the case of symmetric networks. In Sec. IV, we consider the general network case and derive performance guarantees for the Greedy, Randomized and Max-Weight policies. In Sec. V, we establish that the AoI minimization problem is indexable and obtain the Whittle's Index in closed-form. Numerical results are presented in Sec. VI. The paper is concluded in Sec. VII.

II. System Model

Consider a single-hop wireless network with a base station (BS) sending time-sensitive information to M clients. Let the time be slotted, with T consecutive slots forming a frame. At the beginning of every frame, the BS generates one packet per client $i \in \{1, 2, \cdots, M\}$. Those new packets replace any undelivered packets from the previous frame. Denote the frame index by the positive integer k. Packets are periodically generated at every frame k for each client i, thus, each packet can be unequivocally identified by the tuple (k, i).

Let $n \in \{1, \cdots, T\}$ be the index of the slot within a frame. A slot is identified by the tuple (k, n). In a slot, the BS transmits a packet to a selected client i over the wireless channel. The packet is successfully delivered to client i with probability $p_i \in (0, 1]$ and a transmission error occurs with probability $1 - p_i$. The probability of successful transmission p_i is fixed in time, but may differ across clients. The client sends a feedback signal to the BS after every transmission. The feedback (success / failure) reaches the BS instantaneously and without errors.

The transmission scheduling policies considered in this paper are non-anticipative, i.e. policies that do not use future knowledge in selecting clients. Let Π be the class of non-anticipative policies and $\pi \in \Pi$ be an arbitrary admissible policy. In a slot (k, n), policy π can either idle or select a client with an undelivered packet. Clients that have already received their packet by slot (k, n) can only be selected in the next frame $k+1$. Scheduling policies attempt to minimize the expected weighted sum AoI of the clients in the network. Next, we discuss this performance metric.

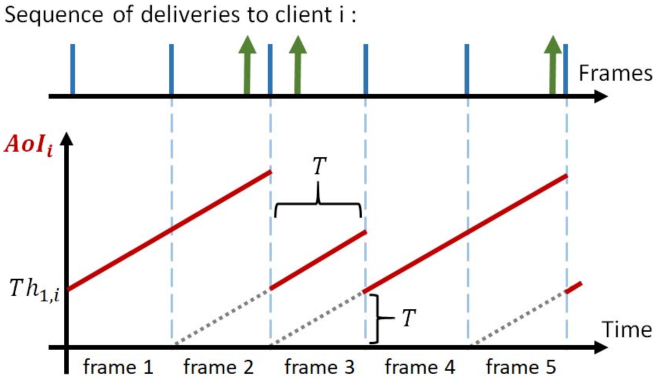

Fig. 2. On the top, a sample sequence of deliveries to client i during five frames. The upward arrows represent the times of packet deliveries. On the bottom, the associated evolution of the AoI_i.

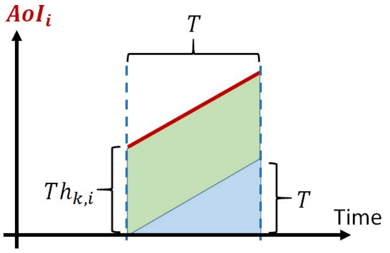

Fig. 3. Area under AoI_i during any frame k in terms of $h_{k,i}$ and T.

A. Age of Information Formulation

Prior to introducing the expected weighted sum AoI, we characterize the Age of Information of a single client in the context of our system model. Let AoI_i be the positive real number that represents the Age of Information of client i. The AoI_i increases linearly in time when there is no delivery of packets to client i. At the end of the frame in which a delivery occurs, the AoI_i is updated to T. In Fig. 2, the evolution of AoI_i is illustrated for a given sample sequence of deliveries to client i.

In Fig. 3, the AoI_i is shown in detail. Let $\hat{s}_k$ denote the set of clients that successfully received packets during frame k and let the positive integer $h_{k,i}$ represent the number of frames since the last delivery[1] to client i. At the beginning of frame $k+1$, the value of $h_{k,i}$ is updated as follows

$$h_{k+1,i} = \begin{cases} h_{k,i} + 1, & \text{if } i \notin \hat{s}_k; \\ 1, & \text{if } i \in \hat{s}_k. \end{cases} \tag{1}$$

As can be seen in Fig. 3, during frame k the area under the AoI_i curve can be divided into a triangle of area $T^2/2$ and a parallelogram of area $h_{k,i}T^2$. This area, averaged over time, captures the average Age of Information associated with client i. A network-wide metric for measuring the freshness of

[1] A similar parameter, denoted Time-Since-Last-Service, is studied in the context of service regularity in [30], [31], [33], and [36].

TABLE II
DESCRIPTION OF KEY NOTATION

M	number of clients. Client index is $i \in \{1,2,\cdots,M\}$
K	number of frames. Frame index is $k \in \{1,2,\cdots,K\}$
T	number of slots in a frame. Slot index is $n \in \{1,2,\cdots,T\}$
p_i	probability of successful transmission to client i
π	admissible non-anticipative scheduling policy
AoI_i	Age of Information associated with client i
$h_{k,i}$	number of frames since the last packet delivery to client i
$\hat{s}_k$	set of clients that received a packet during frame k
α_i	weight of client i. Represents the relative importance of AoI_i
$\mathbb{E}[J_K^\pi]$	objective function that represents the performance of policy π
L_B	Lower Bound on $\mathbb{E}[J_K^\pi]$ for any admissible policy π
U_B^π	Upper Bound on $\mathbb{E}[J_K^\pi]$ for a particular policy π
ρ^π	AoI guarantee associated with policy π
$D_i(K)$	number of packet deliveries to client i up to frame K
$A_i(K)$	number of packet transmissions to client i up to frame K
$I_i[m]$	number of frames between consecutive deliveries to client i
R_i	number of frames remaining after the last delivery to client i
$\bar{\mathbb{M}}[.]$	operator that calculates the sample mean of a set of values
$\bar{\mathbb{V}}[.]$	operator that calculates the sample variance of a set of values

the information is the Expected Weighted Sum AoI, namely

$$\begin{aligned}\text{EWSAoI} &= \frac{1}{KTM}\mathbb{E}\left[\sum_{k=1}^{K}\sum_{i=1}^{M}\alpha_i\left(\frac{T^2}{2}+T^2h_{k,i}\right)\Bigg|\,\vec{h}_1\right]\\ &= \frac{T}{2M}\sum_{i=1}^{M}\alpha_i+\frac{T}{KM}\mathbb{E}\left[\sum_{k=1}^{K}\sum_{i=1}^{M}\alpha_i h_{k,i}\,\Bigg|\vec{h}_1\right], \end{aligned}\quad (2)$$

where α_i is the positive real value that denotes the client's weight and the vector $\vec{h}_1 = [h_{1,1},\cdots,h_{1,M}]^T$ represents the initial values of $h_{k,i}$ in (1). For notation simplicity, we omit $\vec{h}_1$ hereafter. Manipulating the expression of EWSAoI gives us the objective function

$$\min_{\pi\in\Pi}\mathbb{E}\left[J_K^\pi\right], \text{ where } J_K^\pi = \frac{1}{KM}\sum_{k=1}^{K}\sum_{i=1}^{M}\alpha_i\, h_{k,i}^\pi, \quad (3)$$

where (3) is obtained by subtracting the constant terms from (2) and dividing the result by T. As can be seen by the relationship between (2) and (3), the scheduling policy that minimizes $\mathbb{E}\left[J_K^\pi\right]$ is the same policy that minimizes EWSAoI. Henceforth in this paper, we refer to this policy as *AoI-optimal*. With the definitions of AoI[2] and objective function presented, in the next section we introduce the Greedy policy. Table II summarizes key notation.

III. Optimality of Greedy

In this section, we introduce the Greedy policy and show that it minimizes the AoI of the finite-horizon scheduling problem described in Sec. II under some conditions on the underlying network. The Greedy policy is defined next.

Greedy policy schedules in each slot (k,n) a transmission to the client with highest value of $h_{k,i}$ that has an undelivered packet, with ties being broken arbitrarily.

Denote the Greedy policy as G. Observe that Greedy is non-anticipative and work-conserving, i.e. it only idles after all packets have been delivered during frame k. Next, we discuss a few properties of the Greedy policy that lead to the optimality result in Theorem 5.

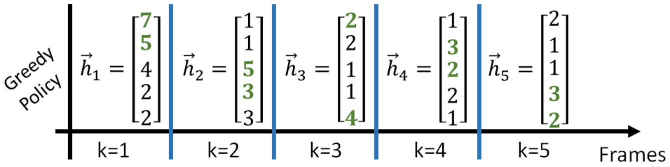
Greedy Policy
$\vec{h}_1 = [7\ 5\ 4\ 2\ 2]^T$
$\vec{h}_2 = [1\ 1\ 5\ 3\ 3]^T$
$\vec{h}_3 = [2\ 2\ 1\ 1\ 4]^T$
$\vec{h}_4 = [1\ 3\ 2\ 2\ 1]^T$
$\vec{h}_5 = [2\ 1\ 1\ 3\ 2]^T$
k=1
k=2
k=3
k=4
k=5
Frames

Fig. 4. Evolution of $\vec{h}_k$ when the Greedy policy is employed in a network with $M = 5$ clients, $T = 2$ slots per frame, error-free channels, $p_i = 1, \forall i$, and $\vec{h}_1 = [7\ 5\ 4\ 2\ 2]^T$. In each frame, the Greedy policy transmits packets of two clients. The elements of $\vec{h}_k$ associated with the clients that received a packet during frame k are depicted in bold green. All elements in $\vec{h}_k$ change according to (1): green elements are updated to 1 while black elements are incremented by 1. In this figure, the Round Robin pattern is evident.

Remark 1: The Greedy policy switches scheduling decisions only after a successful packet delivery.

In slot (k,n), Greedy selects client $i = \arg\max_j\{h_{k,j}\}$ from the set of clients with an undelivered packet. Assume that this packet transmission fails and the subsequent slot is in the same frame k. Since $\vec{h}_k$ remains unchanged and client i still has an undelivered packet, the Greedy policy selects the same client i. Alternatively, if the next slot is in frame $k+1$, then $\vec{h}_{k+1,i}$ evolves according to (1) and Greedy selects $\arg\max_j\{h_{k+1,j}\}$ from the set of all clients. It follows from (1) that client i is selected again. Hence, the Greedy policy selects the same client i, uninterruptedly, until its packet is delivered.

Lemma 2 (Round Robin): Without loss of generality, reorder the client index i in descending order of $\vec{h}_1$, with client 1 having the highest $h_{1,i}$ and client M the lowest $h_{1,i}$. The Greedy policy ***delivers*** *packets according to the index sequence $(1,2,\cdots,M,1,2,\cdots)$ until the end of the time-horizon K, i.e. Greedy follows a Round Robin pattern.*

The proof of Lemma 2 is in Appendix. Together, Remark 1 and Lemma 2, provide a complete description of the behavior of Greedy. Consider a network with $\vec{h}_1$ reordered as in Lemma 2, the Greedy policy schedules client 1, repeatedly, until one packet is delivered, then it schedules client 2, repeatedly, until one packet is delivered, and so on, following the Round Robin pattern until the end of the time-horizon. The Greedy policy only idles when all M packets are delivered in the same frame. Figure 4 illustrates a sequence of scheduling decisions of Greedy in a network with error-free channels.

Corollary 3 (Steady-State of Greedy for Error-Free Channels): Consider a network with error-free channels, $p_i = 1, \forall i$. The Greedy policy drives this network to a steady-state in which the sum of the elements of $\vec{h}_k$ is constant. Let $m_1 \in \mathbb{N}$ and $m_2 \in \{0,1,\cdots,T-1\}$ be the quotient and remainder of the division of M by T, namely $M = m_1T + m_2$. The steady-state is achieved at the beginning of frame $k = m_1 + 2$ and the sum of $\vec{h}_k$ is given by

$$\sum_{i=1}^{M} h_{k,i} = \frac{Tm_1\left(m_1+1\right)}{2} + m_2(m_1+1). \quad (4)$$

Corollary 3 follows directly from the proof of Lemma 2 in Appendix. The sum in (4) comes from the expression of

[2] For ease of exposition, in this paper, the value of AoI_i is updated at the beginning of the frame that follows a successful transmission to client i, rather than immediately after the successful transmission. This update mechanism simplifies the problem while maintaining the features of interest.

$\vec{h}_k$ in (61). Notice that (4) is independent of the initial $\vec{h}_1$. Figure 4 represents a network with $M = 5$, $T = 2$, $m_1 = 2$ and $m_2 = 1$. Thus, according to Corollary 3, the steady-state is achieved in frame $k = 4$ and the sum of the elements of $\vec{h}_k$ is 9 for $k \geq 4$. Those values can be easily verified in Fig. 4.

In Theorem 5, we establish that Greedy is AoI-optimal when the underlying network is *symmetric*, namely all clients have the same channel reliability $p_i = p \in (0, 1]$ and weight $\alpha_i = \alpha \geq 0$. Prior to the main result, we establish in Lemma 4 that Greedy is AoI-optimal for a symmetric network with error-free channels.

Lemma 4 (Optimality of Greedy for Error-Free Channels): Consider a symmetric network with error-free channels $p_i = 1$ and weights $\alpha_i = \alpha > 0, \forall i$. Among the class of admissible policies Π, the Greedy policy attains the minimum sum AoI (2)*, namely*

$$J_K^G \leq J_K^\pi, \forall \pi \in \Pi. \tag{5}$$

The proof of Lemma 4 is in Appendix B of the supplementary material. Intuitively, Greedy minimizes $\sum_{i=1}^{M} h_{k,i}$ by reducing the highest elements of $\vec{h}_k$ to *unity* at every frame. Together, Lemma 4 and Corollary 3 show that, when channels are error-free, Greedy drives the network to a steady-state (4) that is AoI-optimal. Next, we use the result in Lemma 4 to show that the Greedy policy is optimal for any symmetric network.

Theorem 5 (Optimality of Greedy): Consider a symmetric network with channel reliabilities $p_i = p \in (0, 1]$ and weights $\alpha_i = \alpha > 0, \forall i$. Among the class of admissible policies Π, the Greedy policy attains the minimum expected sum AoI (2)*, namely $G = \arg\min_{\pi \in \Pi} \mathbb{E}\left[J_K^\pi\right]$.*

To show that the Greedy policy minimizes the AoI of any symmetric network, we generalize Lemma 4 using a *stochastic dominance* argument [37] that compares the evolution of $\vec{h}_k$ when Greedy is employed to that when an arbitrary policy π is employed. The proof of Theorem 5 is in Appendix C of the supplementary material.

Selecting the client with an undelivered packet and highest value of $h_{k,i}$ in every slot is AoI-optimal for every symmetric network. For general networks, with clients possibly having different channel reliabilities p_i and weights α_i, scheduling decisions based exclusively on $\vec{h}_k$ may not be AoI-optimal. In the next section, we develop three low-complexity scheduling policies and derive performance guarantees for every policy in the context of general networks.

IV. Age of Information Guarantees

One possible approach for finding a policy that minimizes the EWSAoI is to optimize the objective function in (3) using Dynamic Programming [38]. A negative aspect of this approach is that evaluating the optimal scheduling decision for each state of the network can be computationally demanding, especially for networks with a large number of clients.[3] To overcome this problem, known as the curse of dimensionality, and gain insight into the minimization of the Age of Information, we consider four low-complexity scheduling policies, namely Greedy, Randomized, Max-Weight and Whittle's Index policies, and derive performance guarantees for each of them.

For a given network setup (M, K, T, p_i, α_i), the performance of an arbitrary admissible policy $\pi \in \Pi$ is given by $\mathbb{E}\left[J_K^\pi\right]$ from (3) and the optimal performance is $\mathbb{E}\left[J^*\right] = \min_{\eta \in \Pi} \mathbb{E}\left[J_K^\eta\right]$. Ideally, when expressions for $\mathbb{E}\left[J_K^\pi\right]$ and $\mathbb{E}\left[J^*\right]$ are available, we define the optimality ratio[4] $\mathbb{E}\left[J_K^\pi\right]/\mathbb{E}\left[J^*\right]$ and say that policy π is $(\mathbb{E}\left[J_K^\pi\right]/\mathbb{E}\left[J^*\right])$-optimal. Naturally, the closer the optimality ratio is to unity, the better is the performance of policy π. Alternatively, when expressions for $\mathbb{E}\left[J_K^\pi\right]$ and $\mathbb{E}\left[J^*\right]$ are not available, we define the ratio

$$\rho^\pi := \frac{U_B^\pi}{L_B}, \tag{6}$$

where L_B is a lower bound to the AoI-optimal performance and U_B^π is an upper bound to the performance of policy π. It follows from the inequality $L_B \leq \mathbb{E}\left[J^*\right] \leq \mathbb{E}\left[J_K^\pi\right] \leq U_B^\pi$ that $\mathbb{E}\left[J_K^\pi\right]/\mathbb{E}\left[J^*\right] \leq \rho^\pi$ and we can say that policy π is ρ^π-optimal.

Next, we obtain a lower bound L_B that is used for deriving performance guarantees ρ^π for the four low-complexity scheduling policies of interest. Henceforth in this section, *we consider the infinite-horizon problem* where $K \to \infty$. The focus on the long-term behavior of the system allows us to derive simpler and more insightful performance guarantees.

A. Universal Lower Bound

In this section, we find a lower bound to the solution of the objective function in (3).

Theorem 6 (Lower Bound): For a given network setup, we have $L_B \leq \lim_{K\to\infty} \mathbb{E}\left[J_K^\pi\right]$, $\forall \pi \in \Pi$, where

$$L_B = \frac{1}{2MT}\left(\sum_{i=1}^{M}\sqrt{\frac{\alpha_i}{p_i}}\right)^2 + \frac{1}{2M}\sum_{i=1}^{M}\alpha_i. \tag{7}$$

Proof: First, we use a sample path argument to characterize the evolution of $\vec{h}_k$ over time. Then, we derive an expression for the objective function of the infinite-horizon problem, namely $\lim_{K\to\infty} J_K^\pi$, and manipulate this expression to obtain L_B in (7). Fatou's lemma is employed to establish the result in Theorem 6.

Consider a sample path $\omega \in \Omega$ associated with a scheduling policy $\pi \in \Pi$ and a finite time-horizon K. For this sample path, let $D_i(K)$ be the total number of packets delivered to client i up to and including frame K, let $I_i[m]$ be the number of frames between the $(m-1)$th and mth deliveries to client i, i.e. the inter-delivery times of client i, and let R_i be the number of frames remaining after the last packet delivery to the same client. Then, the time-horizon can be written as follows

$$K = \sum_{m=1}^{D_i(K)} I_i[m] + R_i, \quad \forall i \in \{1, 2, \cdots, M\}. \tag{8}$$

[3] Vector $\vec{h}_k = [h_{k,1}, \cdots, h_{k,M}]^T$ is part of the state space of the network. Since each element $h_{k,i}$ can take at least k different values, $h_{k,i} \in \{1, 2, \cdots, k\}$, the set of possible values of $\vec{h}_k$ has cardinality at least k^M. In each slot (k, n) and for every possible network state, the Dynamic Program finds the scheduling decision that minimizes the cost-to-go function associated with the objective in (3). For a time-horizon of K frames, this amounts to $\mathcal{O}(TMK^M)$ operations.

[4] Optimality Ratio is also known as Approximation Ratio.

The evolution of $h_{k,i}$ is well-defined in each of the time intervals $I_i[m]$ and R_i. During the frames associated with the interval $I_i[m]$, the parameter $h_{k,i}$ evolves as $1, 2, \cdots, I_i[m]$. During the frames associated with the interval R_i, the value of $h_{k,i}$ evolves as $1, 2, \cdots, R_i$. Hence, the objective function in (3) can be rewritten as

$$\begin{aligned} J_K^\pi &= \frac{1}{KM}\sum_{k=1}^{K}\sum_{i=1}^{M}\alpha_i h_{k,i} = \frac{1}{M}\sum_{i=1}^{M}\frac{\alpha_i}{K}\left[\sum_{k=1}^{K} h_{k,i}\right] \\ &= \frac{1}{M}\sum_{i=1}^{M}\frac{\alpha_i}{K}\left[\sum_{m=1}^{D_i(K)}\frac{(I_i[m]+1)I_i[m]}{2} + \frac{(R_i+1)R_i}{2}\right] \\ &\overset{(a)}{=} \frac{1}{2M}\sum_{i=1}^{M}\frac{\alpha_i}{K}\left[\sum_{m=1}^{D_i(K)} I_i^2[m] + R_i^2 + K\right] \\ &= \frac{1}{2M}\sum_{i=1}^{M}\alpha_i\left[\frac{D_i(K)}{K}\left(\frac{1}{D_i(K)}\sum_{m=1}^{D_i(K)} I_i^2[m]\right) + \frac{R_i^2}{K} + 1\right], \end{aligned} \tag{9}$$

where (a) uses (8) to substitute the sum of the linear terms $I_i[m]$ and R_i by K.

Now, define the operator $\bar{\mathbb{M}}[.]$ that calculates the sample mean of a set of values. Using this operator, let the sample mean of $I_i[m]$ and $I_i^2[m]$ be

$$\bar{\mathbb{M}}[I_i] = \frac{1}{D_i(K)}\sum_{m=1}^{D_i(K)} I_i[m] \tag{10}$$

$$\bar{\mathbb{M}}[I_i^2] = \frac{1}{D_i(K)}\sum_{m=1}^{D_i(K)} I_i^2[m]. \tag{11}$$

Combining (8) and (10) yields

$$\frac{K}{D_i(K)} = \frac{\sum_{j=1}^{D_i(K)} I_i[j] + R_i}{D_i(K)} = \bar{\mathbb{M}}[I_i] + \frac{R_i}{D_i(K)}. \tag{12}$$

Substituting (11) and (12) into the objective function gives

$$J_K^\pi = \frac{1}{2M}\sum_{i=1}^{M}\alpha_i\left[\left[\bar{\mathbb{M}}[I_i] + \frac{R_i}{D_i(K)}\right]^{-1}\bar{\mathbb{M}}[I_i^2] + \frac{R_i^2}{K} + 1\right], \tag{13}$$

with probability one.

To simplify (13), consider the infinite-horizon problem with $K \to \infty$ and assume that the admissible class Π *does not contain policies that starve clients.*

Definition 7: A policy π starves client i if, with a positive probability, it stops transmitting packets to that client after frame $K' < \infty$.

When π starves client i, the expected number of frames after the last packet delivery is $\mathbb{E}[R_i] \to \infty$ and the objective function $\mathbb{E}[J_K^\pi] \to \infty$. Therefore, policies that starve clients are excluded from the class Π without loss of optimality.

Since policies in Π transmit packets to every client repeatedly and each packet transmission has a positive probability p_i of being delivered, it follows that $I_i[m]$ and R_i are finite with probability one. Thus, in the limit $K \to \infty$, we have $R_i^2/K \to 0$, $D_i(K) \to \infty$ and $R_i/D_i(K) \to 0$. Applying those limits to J_K^π in (13) gives the *objective function of the infinite-horizon AoI problem*

$$\lim_{K\to\infty} J_K^\pi = \frac{1}{2M}\sum_{i=1}^{M}\alpha_i\left[\frac{\bar{\mathbb{M}}[I_i^2]}{\bar{\mathbb{M}}[I_i]} + 1\right] \quad \text{w.p.1.} \tag{14}$$

This insightful expression depicts the relationship between AoI and the moments of the inter-delivery time $I_i[m]$.

Prior to deriving the expression of L_B in (7), we introduce some useful quantities. Define the operator $\bar{\mathbb{V}}[.]$ that calculates the sample variance of a set of values. Let the sample variance of $I_i[m]$ be

$$\bar{\mathbb{V}}[I_i] = \frac{1}{D_i(K)}\sum_{m=1}^{D_i(K)}\left(I_i[m] - \bar{\mathbb{M}}[I_i]\right)^2. \tag{15}$$

Notice that the sample variance is positive valued and $\bar{\mathbb{V}}[I_i] = \bar{\mathbb{M}}[I_i^2] - \left(\bar{\mathbb{M}}[I_i]\right)^2$. Let $A_i(K)$ be the total number of packets *transmitted* to client i up to and including frame K. Any policy π can schedule at most one client per slot, hence

$$\sum_{i=1}^{M} A_i(K) \le KT \quad \text{w.p.1.} \tag{16}$$

Moreover, since every transmission to client i is delivered with the same probability p_i, independently of the outcome of previous transmissions, by the strong law of large numbers

$$\lim_{K\to\infty}\frac{D_i(K)}{A_i(K)} = p_i \quad \text{w.p.1.} \tag{17}$$

With the definitions of $\bar{\mathbb{V}}[I_i]$ and $A_i(K)$, we obtain L_B by manipulating the objective function of the infinite-horizon AoI problem in (14) as follows

$$\begin{aligned} \lim_{K\to\infty} J_K^\pi &= \frac{1}{2M}\sum_{i=1}^{M}\alpha_i\left[\frac{\bar{\mathbb{V}}[I_i]}{\bar{\mathbb{M}}[I_i]} + \bar{\mathbb{M}}[I_i] + 1\right] \\ &\overset{(a)}{\ge} \frac{1}{2M}\sum_{i=1}^{M}\alpha_i\bar{\mathbb{M}}[I_i] + \frac{1}{2M}\sum_{i=1}^{M}\alpha_i \\ &\overset{(b)}{=} \lim_{K\to\infty}\frac{1}{2MT}KT\sum_{i=1}^{M}\frac{\alpha_i}{D_i(K)} + \frac{1}{2M}\sum_{i=1}^{M}\alpha_i \\ &\overset{(c)}{\ge} \lim_{K\to\infty}\frac{1}{2MT}\left(\sum_{j=1}^{M}A_j(K)\right)\left(\sum_{i=1}^{M}\frac{\alpha_i}{D_i(K)}\right) \\ &\quad + \frac{1}{2M}\sum_{i=1}^{M}\alpha_i \\ &\overset{(d)}{\ge} \lim_{K\to\infty}\frac{1}{2MT}\left(\sum_{i=1}^{M}\sqrt{\frac{\alpha_i A_i(K)}{D_i(K)}}\right)^2 + \frac{1}{2M}\sum_{i=1}^{M}\alpha_i \\ &\overset{(e)}{=} \frac{1}{2MT}\left(\sum_{i=1}^{M}\sqrt{\frac{\alpha_i}{p_i}}\right)^2 + \frac{1}{2M}\sum_{i=1}^{M}\alpha_i \quad w.p.1, \end{aligned} \tag{18}$$

where (a) uses the fact that $\bar{\mathbb{V}}[I_i] \ge 0$, (b) uses (12) with $K \to \infty$, (c) uses the inequality in (16), (d) uses Cauchy-Schwarz inequality and (e) uses the equality in (17). Notice that (18) gives the expression for L_B found in (7).

Finally, since J_K^π in (13) is positive for every $\pi \in \Pi$ and for every K, we employ Fatou's lemma to (18) and obtain $\lim_{K\to\infty} \mathbb{E}[J_K^\pi] \geq \mathbb{E}[\lim_{K\to\infty} J_K^\pi] \geq L_B$, establishing the result of the theorem. ■

The sequence of inequalities in (18) that led to $\lim_{K\to\infty} \mathbb{E}[J_K^\pi] \geq L_B$ could have rendered a loose lower bound. However, in the next section, we use L_B to derive a performance guarantee ρ^G for the Greedy policy and show that $\rho^G = 1$ for symmetric networks with large M, i.e. under these conditions the value of L_B is as tight as possible. Furthermore, numerical results in Sec. VI show that the lower bound is also tight in other network configurations. In the upcoming sections, we obtain performance guarantees for four low-complexity scheduling policies: Greedy, Randomized, Max-Weight and Whittle's Index policies.

B. Greedy Policy

In this section, we analyze the Greedy policy introduced in Sec. III and derive a closed-form expression for its performance guarantee ρ^G. The expression for ρ^G depends on the statistics of the set of values $\{1/p_i\}_{i=1}^M$, in particular of its *coefficient of variation*. Let the sample mean and sample variance of $\{1/p_i\}_{i=1}^M$ be

$$\bar{\mathbb{M}}\left[\frac{1}{p_i}\right] = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{p_j}; \tag{19}$$

$$\bar{\mathbb{V}}\left[\frac{1}{p_i}\right] = \frac{1}{M}\sum_{j=1}^{M}\left(\frac{1}{p_j} - \bar{\mathbb{M}}\left[\frac{1}{p_i}\right]\right)^2. \tag{20}$$

Then, the coefficient of variation is given by

$$C_V = \sqrt{\bar{\mathbb{V}}\left[\frac{1}{p_i}\right]}\Bigg/\bar{\mathbb{M}}\left[\frac{1}{p_i}\right]. \tag{21}$$

The coefficient of variation is a measure of how spread out are the values of $1/p_i$. The value of C_V is large when $\{1/p_i\}_{i=1}^M$ are disperse and $C_V = 0$ if and only if $p_i = p$ for all clients.

Theorem 8 (Performance of Greedy): Consider a network (M, T, p_i, α_i) with an infinite time-horizon. The Greedy policy is ρ^G-optimal as $M \to \infty$, where

$$\rho^G = \frac{\bar{\mathbb{M}}[\alpha_i]\,\bar{\mathbb{M}}\left[\frac{1}{p_i}\right]}{\left(\bar{\mathbb{M}}\left[\sqrt{\frac{\alpha_i}{p_i}}\right]\right)^2}\left(1 + \frac{C_V^2}{M}\right), \tag{22}$$

when the sample means and sample variance are well-defined as $M \to \infty$.

The proof of Theorem 8 is in Appendix D of the supplementary material. The expression of ρ^G for finite M can be readily obtained by dividing (87) by (7). Next, we use the performance guarantee in (22) to obtain sufficient conditions for the optimality of the Greedy policy.

Corollary 9: The Greedy policy minimizes the expected sum AoI (2) of any symmetric network with $M \to \infty$.

Proof: Consider two inequalities. (i) Cauchy-Schwarz

$$\left(\frac{1}{M}\sum_{i=1}^{M}\sqrt{\frac{\alpha_i}{p_i}}\right)^2 \leq \left(\frac{1}{M}\sum_{i=1}^{M}\alpha_i\right)\left(\frac{1}{M}\sum_{i=1}^{M}\frac{1}{p_i}\right), \tag{23}$$

and (ii) Positive coefficient of variation: $C_V \geq 0$. It is evident from (22) that $\rho^G = 1$ if and only if both inequalities (i) and (ii) hold with *equality* and this is true if and only if $\alpha_i = \alpha$ and $p_i = p$ for all clients. ■

Theorem 8 provides a closed-form expression for the performance guarantee ρ^G and Corollary 9 shows that, by leveraging the knowledge of $h_{k,i}$, the Greedy policy achieves optimal performance in symmetric networks with $M \to \infty$. Notice that the Greedy policy does not take into account differences in terms of weight α_i and channel reliability p_i. In the next section, we study the class of Stationary Randomized policies which use the knowledge of α_i and p_i but neglect $h_{k,i}$.

C. Stationary Randomized Policy

Consider the class of Stationary Randomized policies in which scheduling decisions are made randomly, according to fixed probabilities. In particular, define the Randomized policy as follows.

Randomized policy selects in each slot (k, n) client i with probability $\beta_i / \sum_{j=1}^M \beta_j$, for every client i and for positive fixed values of $\{\beta_i\}_{i=1}^M$. The BS transmits the packet if the selected client has an undelivered packet and idles otherwise.

Denote the Randomized policy as R. Observe that this policy uses no information from current or past states of the network. Moreover, it is not work-conserving, since the BS can idle when the network still has clients with undelivered packets. Next, we derive a closed-form expression for the performance guarantee ρ^R and find a Randomized policy that is 2-optimal *for all network configurations with $T = 1$* slot per frame.

Theorem 10 (Performance of Randomized): Consider a network (M, T, p_i, α_i) with an infinite time-horizon. The Randomized policy with positive values of $\{\beta_i\}_{i=1}^M$ is ρ^R-optimal, where

$$\rho^R = 2\frac{\left(\sum_{j=1}^{M}\beta_j\sum_{i=1}^{M}\frac{\alpha_i}{p_i\beta_i}\right) + (T-1)\left(\sum_{i=1}^{M}\frac{\alpha_i}{p_i}\right)}{\left(\sum_{i=1}^{M}\sqrt{\frac{\alpha_i}{p_i}}\right)^2 + T\left(\sum_{i=1}^{M}\alpha_i\right)}. \tag{24}$$

Proof: The performance guarantee is defined as $\rho^R = U_B^R / L_B$, where the denominator is the universal lower bound in (7) and the numerator is an upper bound to the objective function, namely $\lim_{K\to\infty}\mathbb{E}[J_K^R] \leq U_B^R$, which is derived in Appendix E of the supplementary material.

Let $d_i(k) \in \{0, 1\}$ be the number of packets delivered to client i during frame k. Notice that

$$\mathbb{E}[d_i(k)] = \mathbb{P}(\text{delivery to client } i \text{ during frame } k). \tag{25}$$

When the Randomized policy is employed, this probability is constant over time, i.e. $\mathbb{E}[d_i(k)] = \mathbb{E}[d_i]$. Moreover, the PMF of the random variable $I_i[m]$ that represents the number of frames between the $(m-1)$th and mth packet deliveries to client i is given by

$$\mathbb{P}(I_i[m] = n) = \mathbb{E}[d_i](1 - \mathbb{E}[d_i])^{n-1}, \tag{26}$$

for $n \in \{1, 2, \cdots\}$ and is *independent of* m.

Clearly, when the Randomized policy is employed, the sequence of packet deliveries is a renewal process with geometric inter-delivery times $I_i[m]$. Thus, using the generalization of the elementary renewal theorem for renewal-reward processes [39, Sec. 5.7] yields

$$\lim_{K\to\infty}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[h_{k,i}]=\frac{\mathbb{E}[I_i[m]^2]}{2\mathbb{E}[I_i[m]]}+\frac{1}{2}=\frac{1}{\mathbb{E}[d_i]},\tag{27}$$

and substituting (27) into the objective function (3) gives

$$\lim_{K\to\infty}\mathbb{E}\left[J_K^R\right]=\frac{1}{M}\sum_{i=1}^{M}\frac{\alpha_i}{\mathbb{E}\left[d_i\right]}.\tag{28}$$

For simplicity of exposition, we consider the case $T=1$ slot per frame. The derivation of the performance guarantee ρ^R for general T is in Appendix E. When $T=1$, packets are always available for transmission and the BS selects one client per frame. Hence, the probability of delivering a packet to client i during frame k is

$$\mathbb{E}\left[d_i\right]=\frac{\beta_i}{\sum_{j=1}^{M}\beta_j}p_i.\tag{29}$$

Substituting (29) into (28) gives

$$\lim_{K\to\infty}\mathbb{E}\left[J_K^R\right]=\frac{1}{M}\sum_{j=1}^{M}\beta_j\sum_{i=1}^{M}\frac{\alpha_i}{p_i\beta_i}=U_B^R.\tag{30}$$

Finally, dividing (30) by the lower bound in (7) gives the performance guarantee ρ^R in (24) for $T=1$. ■

Corollary 11: The Randomized policy with $\beta_i=\sqrt{\alpha_i/p_i},\forall i$, has $\rho^R<2$ for all networks with $T=1$ slot per frame.

Proof: The assignment $\beta_i=\sqrt{\alpha_i/p_i},\forall i\in\{1,\cdots,M\}$ is the necessary (and sufficient) condition for the Cauchy-Schwarz inequality

$$\left(\sum_{i=1}^{M}\sqrt{\frac{\alpha_i}{p_i}}\right)^2\leq\left(\sum_{j=1}^{M}\beta_j\right)\left(\sum_{i=1}^{M}\frac{\alpha_i}{\beta_ip_i}\right),\tag{31}$$

to hold with equality. Applying this condition to (24) for $T=1$ results in $\rho^R<2$. Notice that $\beta_i=\sqrt{\alpha_i/p_i}$ is the assignment which minimizes the RHS of (31) and the expression in (30). ■

Theorem 10 gives an expression for ρ^R and Corollary 11 shows that, by using only the knowledge of α_i and p_i, a Randomized policy can achieve 2-optimal performance in a *wide range of network setups*, in particular all networks with $T=1$ slot per frame. Next, we develop a Max-Weight policy that leverages the knowledge of α_i, p_i and $h_{k,i}$ in making scheduling decisions.

D. Max-Weight Policy

In this section, we use concepts from Lyapunov Optimization [40] to derive a Max-Weight policy. The Max-Weight policy is obtained by minimizing the drift of a Lyapunov Function of the system state at every frame k. Consider the quadratic Lyapunov Function

$$L(\vec{h}_k)=\frac{1}{M}\sum_{i=1}^{M}\alpha_ih_{k,i}^2,\tag{32}$$

and the one-frame Lyapunov Drift

$$\Delta(\vec{h}_k)=\mathbb{E}\left[L(\vec{h}_{k+1})-L(\vec{h}_k)\middle|\vec{h}_k\right].\tag{33}$$

The Lyapunov Function $L(\vec{h}_k)$ depicts how large the AoI of the clients in the network during frame k is, while the Lyapunov Drift $\Delta(\vec{h}_k)$ represents the growth of $L(\vec{h}_k)$ from one frame to the next. Intuitively, by minimizing the drift, the Max-Weight policy reduces the value of $L(\vec{h}_k)$ and, consequently, keeps the AoI of the clients low.

To find the policy that minimizes the one-frame drift $\Delta(\vec{h}_k)$, we first need to analyze the RHS of (33). Consider frame k with a fixed vector $\vec{h}_k$ and a policy π making scheduling decisions throughout the T slots of this frame. Recall that $d_i^\pi(k)\in\{0,1\}$ represents the number of packets delivered to client i during frame k when policy π is employed. An alternative way to represent the evolution of $h_{k,i}$ defined in (1) is

$$h_{k+1,i}=d_i^\pi(k)+(h_{k,i}+1)[1-d_i^\pi(k)].\tag{34}$$

Applying (34) into the conditional expectation of $h_{k+1,i}^2$ yields

$$\mathbb{E}\left[h_{k+1,i}^2-h_{k,i}^2|\vec{h}_k\right]=-\mathbb{E}\left[d_i^\pi(k)|\vec{h}_k\right]h_{k,i}(h_{k,i}+2)\\+2h_{k,i}+1.\tag{35}$$

Substituting (32) into (33) and then using (35) gives the following expression for the Lyapunov Drift

$$\Delta(\vec{h}_k)=-\frac{1}{M}\sum_{i=1}^{M}\mathbb{E}\left[d_i^\pi(k)|\vec{h}_k\right]\alpha_ih_{k,i}(h_{k,i}+2)\\+\frac{2}{M}\sum_{i=1}^{M}\alpha_ih_{k,i}+\frac{1}{M}\sum_{i=1}^{M}\alpha_i.\tag{36}$$

Observe that the scheduling policy π only affects the first term on the RHS of (36). Define the weight function $G_i(h_{k,i})=\alpha_ih_{k,i}(h_{k,i}+2)$. During frame k, the scheduling policy that maximizes the sum $\sum_{i=1}^{M}\mathbb{E}\left[d_i^\pi(k)|\vec{h}_k\right]G_i(h_{k,i})$ also minimizes $\Delta(\vec{h}_k)$. Notice that $\mathbb{E}\left[d_i^\pi(k)|\vec{h}_k\right]$ represents the expected throughput of client i during frame k. The class of policies that maximize the *expected weighted sum throughput in a frame* was studied in [27] and [41]. According to [41, eq. (2)], to maximize $\sum_{i=1}^{M}\mathbb{E}\left[d_i^\pi(k)|\vec{h}_k\right]G_i(h_{k,i})$, the scheduling policy must myopically select the client with an undelivered packet and highest value of $p_iG_i(h_{k,i})$ in every slot of frame k. Hence, the Max-Weight policy is defined as follows.

Max-Weight policy schedules in each slot (k,n) a transmission to the client with highest value of $p_i\alpha_ih_{k,i}(h_{k,i}+2)$ that has an undelivered packet, with ties being broken arbitrarily.

Denote the Max-Weight policy as MW. Observe that when $\alpha_i=\alpha$ and $p_i=p$, prioritizing according to $p_i\alpha_ih_{k,i}(h_{k,i}+2)$

is identical to prioritizing according to $h_{k,i}$, i.e. Max-Weight is identical to Greedy. Thus, from Theorem 5 (Optimality of Greedy), we conclude that Max-Weight is AoI-optimal for symmetric networks. For general networks, we derive the performance guarantee ρ^{MW} for the Max-Weight policy.

Theorem 12 (Performance of Max-Weight): Consider a network (M, T, p_i, α_i) with an infinite time-horizon. The Max-Weight policy is ρ^{MW}-optimal, where

$$\rho^{MW} = 4\frac{\left(\sum_{i=1}^{M}\sqrt{\frac{\alpha_i}{p_i}}\right)^2 + (T-1)\sum_{i=1}^{M}\frac{\alpha_i}{p_i}}{\left(\sum_{i=1}^{M}\sqrt{\frac{\alpha_i}{p_i}}\right)^2 + T\left(\sum_{i=1}^{M}\alpha_i\right)}. \tag{37}$$

The proof of Theorem 12 is in Appendix F of the supplementary material. In contrast to the Greedy and Randomized policies, the Max-Weight policy uses all available information, namely p_i, α_i and $h_{k,i}$, in making scheduling decisions. As expected, numerical results in Sec. VI demonstrate that Max-Weight outperforms both Greedy and Randomized in every network setup simulated. In fact, the performance of Max-Weight is comparable to the optimal performance computed using Dynamic Programming. However, by comparing the performance guarantee ρ^{MW} in (37) with ρ^G and ρ^R, it might seem that Max-Weight does not provide better performance. The reason for this is the challenge to obtain a tight performance upper bound for Max-Weight. As opposed to Greedy and Randomized, the Max-Weight policy cannot be evaluated using Renewal Theory and it does not have properties that simplify the analysis, such as packets being delivered following a Round Robin pattern or clients being selected according to fixed probabilities. Next, we consider the AoI minimization problem from a different perspective and propose an Index policy [42], also known as Whittle's Index policy. This policy is surprisingly similar to the Max-Weight policy and also yields a strong performance.

V. Whittle's Index Policy

Whittle's Index policy is the optimal solution to a relaxation of the Restless Multi-Armed Bandit (RMAB) problem. This low-complexity heuristic policy has been extensively used in the literature [29], [32], [43] and is known to have a strong performance in a range of applications [44], [45]. The challenge associated with this approach is that the Index policy is only defined for problems that are *indexable*, a condition which is often difficult to establish.

To develop the Whittle's Index policy, the AoI minimization problem is transformed into a relaxed RMAB problem. The first step is to note that each client in the AoI problem evolves as a restless bandit. Thus, the AoI problem can be posed as a RMAB problem. The second step is to consider the relaxed version of the RMAB problem, called the Decoupled Model, in which clients are examined separately. The Decoupled Model associated with each client i adheres to the network model with $M = 1$, except for the addition of a *service charge*. The service charge is a fixed cost per transmission $C > 0$ that is incurred by the network every time the BS transmits a packet. The last step is to solve the Decoupled Model. This solution lays the foundation for the design of the Index policy. Next, we formulate and solve the Decoupled Model, establish that the AoI problem is indexable and derive the Whittle's Index policy. A detailed introduction to the Whittle's Index policy can be found in [42] and [46].

A. Decoupled Model

The Decoupled Model is formulated as a Dynamic Program (DP). For presenting the cost-to-go function, which is central to the DP, we first introduce the state, control, transition and objective of the model. Then, using the expression of the cost-to-go, we establish in Proposition 13 a key property of the Decoupled Model which is used in the characterization of its optimal scheduling policy. Since the Decoupled Model considers only a single client, hereafter in this section, we omit the client index i.

Consider the network model from Sec. II with $M = 1$ client. Recall that at the beginning of every frame, the BS generates a new packet that replaces any undelivered packet from previous frame. Let $s_{k,n}$ represent the delivery status of this new packet at the beginning of slot (k, n). If the packet has been successfully delivered to the client by the beginning of slot (k, n), then $s_{k,n} = 1$, and if the packet is still undelivered, $s_{k,n} = 0$. The tuple $(s_{k,n}, h_k)$ depicts the system state, for it provides a complete characterization of the network at slot (k, n).

Denote by $u_{k,n}$ the scheduling decision in time slot (k, n). This quantity is equal to 1 if the BS transmits the packet in slot (k, n), and $u_{k,n} = 0$ otherwise. Since the BS can only transmit undelivered packets, if $s_{k,n} = 1$, then the decision must be to idle $u_{k,n} = 0$.

State transitions are different at frame boundaries and within frames. At the boundary between frames $k-1$ and k, namely, in the transition from slot $(k-1, T)$ to slot $(k, 1)$, each component of the system state $(s_{k,n}, h_k)$ evolves in a distinct way. Since the BS generates a new packet at the beginning of slot $(k, 1)$, we have $s_{k,1} = 0$ for every frame k. Whereas, the evolution of h_k is divided into two cases: i) case $u_{k-1,T} = 1$, when the BS transmits the packet during slot $(k-1, T)$, the value of h_k depends on the feedback signal, as follows

$$P(h_k = h_{k-1} + 1 | h_{k-1}) = 1 - p; \quad \text{[failure]} \tag{38}$$

$$P(h_k = 1 | h_{k-1}) = p; \quad \text{[success]} \tag{39}$$

and ii) case $u_{k-1,T} = 0$, when the BS idles, the transition is deterministic

$$P(h_k = h_{k-1} + 1 | h_{k-1}) = 1, \quad \text{if } s_{k-1,T} = 0; \tag{40}$$

$$P(h_k = 1 | h_{k-1}) = 1, \quad \text{if } s_{k-1,T} = 1. \tag{41}$$

For state transitions that occur within the same frame, the quantity h_k remains fixed and $s_{k,n}$ evolves according to the scheduling decisions and feedback signals. If the BS idles during slot $(k, n-1)$, the delivery status of the packet does not change, thus

$$P(s_{k,n} = s_{k,n-1} | s_{k,n-1}) = 1. \tag{42}$$

If the BS transmits during slot $(k, n-1)$, the value of $s_{k,n}$ depends upon the outcome of the transmission, as given by

$$P(s_{k,n} = 0 | s_{k,n-1}) = 1 - p; \quad \text{[failure]} \tag{43}$$
$$P(s_{k,n} = 1 | s_{k,n-1}) = p. \quad \text{[success]} \tag{44}$$

The last concept to be discussed prior to the cost-to-go function is the objective. The objective function of the Decoupled Model, $\mathcal{J}_K^\pi$, is analogous to J_K^π in (3), except that it represents a single client, introduces the service charge C and evolves in slot increments (instead of frame increments). The expression for the objective function is given by

$$\min_{\pi \in \Pi} \mathbb{E}\left[\mathcal{J}_K^\pi\right],$$
$$\text{where } \mathcal{J}_K^\pi = \frac{1}{KT} \sum_{k=1}^{K} \sum_{n=1}^{T} \left(\alpha\, h_k + C\, u_{k,n}\right). \tag{45}$$

The cost-to-go function $\mathcal{J}_{k,n}(s_{k,n}, h_k)$ associated with the optimization problem in (45) has two forms. For the *last* slot of any frame k, namely slot (k, T), the cost-to-go is expressed as

$$\mathcal{J}_{k,T}(s_{k,T}, h_k) = \alpha h_k + \min_{u_{k,T} \in \{0,1\}} \left\{ C\, u_{k,T} + \mathbb{E}[\mathcal{J}_{k+1,1}(0, h_{k+1})] \right\}, \tag{46}$$

and for slots *other than the last*, we have

$$\mathcal{J}_{k,n}(s_{k,n}, h_k) = \alpha h_k + \min_{u_{k,n} \in \{0,1\}} \left\{ C\, u_{k,n} + \mathbb{E}[\mathcal{J}_{k,n+1}(s_{k,n+1}, h_k)] \right\}. \tag{47}$$

Given a network setup $(K, T, p, \alpha, h_1, C)$, it is possible to use backward induction on (46) and (47) to compute the optimal scheduling policy π^* for the Decoupled Model. However, for the purpose of designing the Index policy, it is not sufficient to provide an algorithm that computes the optimal policy. The Index policy is based on a complete characterization of π^*. Proposition 13 provides a key feature of the optimal scheduling policy which is used in its characterization.

Proposition 13: Consider the Decoupled Model and its optimal scheduling policy π^*. During any frame k, the optimal policy either: (i) idles in every slot; or (ii) transmits until the packet is delivered or the frame ends.

Proof: The proof follows from the analysis of the backward induction algorithm on (46) and (47). For this proof, we assume that the algorithm has been running and that the values of $\mathcal{J}_{k+1,1}(s_{k+1,1}, h_{k+1})$ for all possible system states are known. The proof is centered around the backward induction during frame k and for a fixed value of h_k.

First, we analyze the (trivial) case in which the packet has already been delivered by the beginning of slot (k, n), i.e. $s_{k,n} = 1$. In this case, the optimal scheduling policy always idles.

For the more interesting case of an undelivered packet, we start by analyzing the last slot of the frame, namely slot (k, T). It follows from the cost-to-go in (46) that the optimal scheduling decision $u_{k,T}^*$ depends only on the expression

$$C - p\left[\mathcal{J}_{k+1,1}(0, h_k + 1) - \mathcal{J}_{k+1,1}(0, 1)\right]. \tag{48}$$

The optimal policy idles in slot (k, T) if (48) is non-negative and transmits if (48) is negative. By analyzing the cost-to-go function in (47), which is associated with the optimal scheduling decisions in the remaining slots of frame k, it is possible to use mathematical induction to establish that:

- if it is optimal to transmit in slot $(k, n+1)$, then it is also optimal to transmit in slot (k, n); and
- if it is optimal to idle in slot $(k, n+1)$, then it is also optimal to idle in slot (k, n).

We conclude that if (48) is non-negative, the optimal policy idles in every slot of frame k, and if (48) is negative, the optimal policy transmits until the packet is delivered or until frame k ends. ■

Let $\Gamma \subset \Pi$ be the subclass of all scheduling policies that satisfy Proposition 13. Since the optimal policy is such that $\pi^* \in \Gamma$, we can reduce the scope of the Decoupled Model to policies in Γ without loss of optimality. In the following section, we redefine the Decoupled Model so that scheduling decisions are made only once per frame, rather than once per slot. This new model is denoted Frame-Based Decoupled Model.

B. Frame-Based Decoupled Model

Denote by u_k the scheduling decision at the beginning of frame k. We let $u_k = 0$ if the BS idles in every slot of frame k and $u_k = 1$ if the BS transmits repeatedly until the packet is delivered or the frame ends.

Since this discrete-time decision problem evolves in frames and every frame begins with $s_{k,1} = 0$, we can fully represent the system state by h_k. State transitions follow the evolution of h_k in (1) and can be divided into two cases: i) case $u_{k-1} = 0$, when the BS idles during frame $k-1$

$$P(h_k = h_{k-1} + 1 | h_{k-1}) = 1, \tag{49}$$

and ii) case $u_{k-1} = 1$, when the BS transmits, the state transition depends on whether the packet was delivered or discarded during frame $k-1$, as follows

$$P(h_k = h_{k-1} + 1 | h_{k-1}) = (1-p)^T; \quad \text{[discarded]} \tag{50}$$
$$P(h_k = 1 | h_{k-1}) = 1 - (1-p)^T. \quad \text{[delivered]} \tag{51}$$

The objective function of the Frame-Based Decoupled Model, $\hat{\mathcal{J}}_K^\pi$, is given by

$$\min_{\pi \in \Gamma} \mathbb{E}\left[\hat{\mathcal{J}}_K^\pi\right], \quad \text{where } \hat{\mathcal{J}}_K^\pi = \frac{1}{KT} \sum_{k=1}^{K} \left(T\alpha h_k + \hat{C} u_k\right), \tag{52}$$

and $\hat{C} = C(1 - (1-p)^T)/p$ is the expected value of the service charge incurred during a frame in which the BS transmits. By construction, the Frame-Based Decoupled Model is equivalent to the Decoupled Model when the optimization is carried over the policies in Γ. Thus, both models have the same optimal scheduling policy $\pi^* \in \Gamma \subset \Pi$. Next, we characterize π^* for the infinite-horizon problem.

Consider the Frame-Based Decoupled Model over an infinite-horizon with $K \to \infty$. The state and control of the system in *steady-state* are denoted h and u, respectively. Then, Bellman equations are given by $S(1) = 0$ and

$$\begin{aligned} S(h) + \lambda T = \min\{&T\alpha h + S(h+1) \\ &\hat{C} + T\alpha h + (1-p)^T S(h+1) \\ &+ (1-(1-p)^T)S(1)\}, \end{aligned} \tag{53}$$

for all $h \in \{1, 2, \cdots\}$, where λ is the optimal average cost and $S(h)$ is the differential cost-to-go function. Notice that the upper part of the minimization in (53) is associated with choosing $u = 0$, i.e. idling in every slot of the frame, and the lower part with $u = 1$, i.e. transmitting until the packet is delivered or the frame ends, with ties being broken in favor to idling. The stationary scheduling policy that solves Bellman equations[5] is given in Proposition 14.

Proposition 14 (Threshold Policy): Consider the Frame-Based Decoupled Model over an infinite-horizon. The stationary scheduling policy π^* that solves Bellman equations (53) *is a threshold policy in which the BS transmits during frames that have $h > H - 1$ and idles when $1 \leq h \leq H - 1$, where the threshold H is given by*

$$H = \left\lfloor 1 - Z + \sqrt{Z^2 + \frac{2C}{pT\alpha}} \right\rceil, \tag{54}$$

and the value of Z is

$$Z = \frac{1}{2} + \frac{(1-p)^T}{(1-(1-p)^T)}. \tag{55}$$

The proof of Proposition 14 is in Appendix G of the supplementary material. Intuitively, we expect that the optimal scheduling decision is to transmit during frames in which h is high (attempting to reduce the value of h) and to idle when h is low (avoiding the service charge $\hat{C}$). Moreover, if the optimal decision is to transmit when the state is $h = H$, it is natural to expect that for all $h \geq H$ the optimal decision is also to transmit. This behavior characterizes a threshold policy. In Appendix G, we demonstrate this behavior and find the minimum integer H for which the optimal decision is to transmit. With the complete characterization of π^* provided in Proposition 14, we have the necessary background to establish indexability and to obtain the Whittle's Index policy for the AoI minimization problem.

C. Indexability and Index Policy

Consider the Decoupled Model and its optimal scheduling policy π^*. Let $\mathcal{P}(C)$ be the set of states h for which it is optimal to idle when the service charge is C, i.e. $\mathcal{P}(C) = \{h \in \mathbb{N} | h < H\}$. Note from (54) that the threshold H is a function of C. The definition of indexability is given next.

Definition 15 (Indexability): The Decoupled Model associated with client i is indexable if $\mathcal{P}(C)$ increases monotonically from $\emptyset$ to the entire state space, $\mathbb{N}$, as the service charge C increases from 0 to $+\infty$. The AoI minimization problem is indexable if the Decoupled Model is indexable for all clients i.

The indexability of the Decoupled Model follows directly from the expression of H in (54). Clearly, the threshold H is monotonically increasing with C. Also, substituting $C = 0$ yields $H = 1$, which implies $\mathcal{P}(C) = \emptyset$, and the limit $C \to +\infty$ gives $H \to +\infty$ and, consequently, $\mathcal{P}(C) = \mathbb{N}$. Since this is true for the Decoupled Model associated with every client i, we conclude that the AoI minimization problem is indexable. Prior to introducing the Index policy, we define the Whittle's Index.

Definition 16 (Index): Consider the Decoupled Model and denote by $C(h)$ the Whittle's Index in state h. Given indexability, $C(h)$ is the infimum service charge C that makes both scheduling decisions (idle, transmit) equally desirable in state h.

The closed-form expression for $C(h)$ comes from the fact that, for both scheduling decisions to be equally desirable in state h, the threshold must be $H = h + 1$. Substituting $H = h + 1$ into (54) and isolating C, gives

$$C(h) = p\alpha h \left[h + \frac{1 + (1-p)^T}{1 - (1-p)^T} \right]. \tag{56}$$

After establishing indexability and finding the closed-form expression for the Whittle's Index, we return to our original problem, with the BS transmitting packets to M clients. Recall that there is no service charge in the original problem. The Whittle's Index policy is described next.

Whittle's Index policy schedules in each slot (k, n) a transmission to the client with highest value of

$$C_i(h_{k,i}) = p_i \alpha_i h_i \left[h_i + \frac{1 + (1-p_i)^T}{1 - (1-p_i)^T} \right], \tag{57}$$

that has an undelivered packet, with ties being broken arbitrarily.

Denote the Whittle's Index policy as WI. By construction, the index $C_i(h_{k,i})$ represents the service charge that the network would be willing to pay in order to transmit a packet to client i during frame k. Intuitively, by selecting the client with highest $C_i(h_{k,i})$, the Whittle's Index policy is transmitting the most valuable packet. Note that the Whittle's Index policy is similar to the Max-Weight policy despite the fact that they were developed using different methods. Both the Whittle's Index and Max-Weight policies have strong performances and both are equivalent to the Greedy policy when the network is symmetric, implying that WI and MW are AoI-optimal when $\alpha_i = \alpha$ and $p_i = p$. Next, we derive the performance guarantee ρ^{WI} for the Whittle's Index policy.

Theorem 17 (Performance of Whittle): Consider a network (M, T, p_i, α_i) with an infinite time-horizon. The Whittle's Index policy is ρ^{WI}-optimal, where

$$\rho^{WI} = 4 \frac{\left(\sum_{i=1}^{M} \sqrt{\frac{\widetilde{\alpha}_{\mathbf{i}}}{p_i}} \right)^2 + (T-1) \sum_{i=1}^{M} \frac{\widetilde{\alpha}_{\mathbf{i}}}{p_i}}{\left(\sum_{i=1}^{M} \sqrt{\frac{\alpha_i}{p_i}} \right)^2 + T \left(\sum_{i=1}^{M} \alpha_i \right)}, \tag{58}$$

[5] In general, Expected Average Cost problems over an infinite-horizon and with *countably infinite state space* are challenging to address. For the Frame-Based Decoupled Model, it can be shown that [38, Proposition 5.6.1] is satisfied under some additional conditions on Γ. The results in [38, Proposition 5.6.1] and Proposition 14 are sufficient to establish the optimality of the stationary scheduling policy π^*.

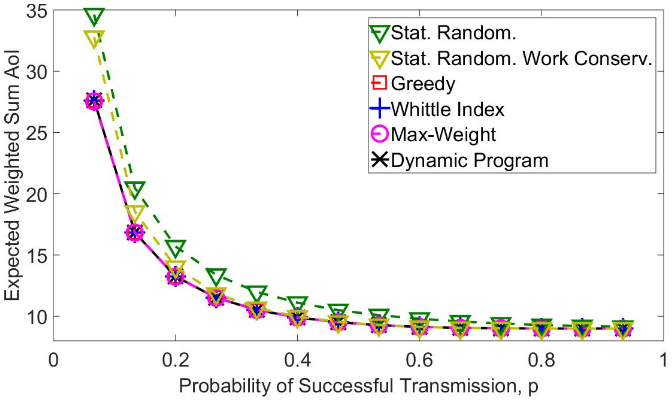

Fig. 5. Two-user symmetric network with $T = 6, K = 150, \alpha_i = 1, p_i = p, \forall i$. The simulation result for each policy and for each value of p is an average over $1,000$ runs.

and

$$\widetilde{\alpha}_{\mathbf{i}} = \frac{\alpha_i}{2}\left(\frac{2}{1-(1-p_i)^T}+1\right)^2. \tag{59}$$

To find the expression for the performance guarantee of the Whittle's Index policy ρ^{WI} in (58), we use similar arguments to the ones for deriving ρ^{MW}. The proof of Theorem 17 is in Appendix H of the supplementary material. Next, we evaluate the performance of the four low-complexity scheduling policies discussed in this paper using MATLAB simulations.

VI. Simulation Results

In this section, we evaluate the performance of the scheduling policies in terms of the Expected Weighted Sum Age of Information in (2). We compare six scheduling policies: i) Greedy policy; ii) Randomized policy with $\beta_i = \sqrt{\alpha_i/p_i}$; iii) Work-Conserving Randomized policy with $\beta_i = \sqrt{\alpha_i/p_i}$; iv) Max-Weight policy; v) Whittle's Index policy and vi) the optimal Dynamic Program. The numerical results associated with the first five policies are *simulations*, while the results associated with the Dynamic Program are *computations* of EWSAoI obtained by applying Value Iteration to the objective function (3). By definition, the Dynamic Program yields the optimal performance. The Work-Conserving Randomized policy is identical to the Randomized policy, except that it does not idle when the selected client has already delivered its packet. In this case, it makes a new selection, using the same probabilities $\beta_i / \sum_{j=1}^{M}\beta_j$.

Figs. 5 and 6 evaluate the scheduling policies in a variety of network settings. In Fig. 5, we consider a two-user *symmetric* network with $T = 6$ slots in a frame, a total of $K = 150$ frames and both clients having the same weight $\alpha_1 = \alpha_2 = 1$ and channel reliability $p_1 = p_2 \in \{1/15, \cdots, 14/15\}$. In Fig. 6, we consider a two-user *non-symmetric* network with $K = 200$, $\alpha_1 = 2$, $\alpha_2 = 1$, $p_1 = 2/3$, $p_2 = 1/7$ and $T \in \{1, \cdots, 10\}$. The initial vector is $\vec{h}_1 = [1, 1, \cdots, 1]^T$ in all simulations.

Figs. 7 displays the performance of the scheduling policies for larger networks. Due to the high computation complexity associated with the Dynamic Program, we show the Lower Bound L_B from (7) instead. We consider a network with an increasing number of clients $M \in \{5, 10, \cdots, 45, 50\}$, $T = 3$ slots in a frame, a total of $K = 50,000$ frames, channel reliability $p_i = i/M$, $\forall i \in \{1, 2, \cdots, M\}$ and all clients having the same weight $\alpha_i = 1$.

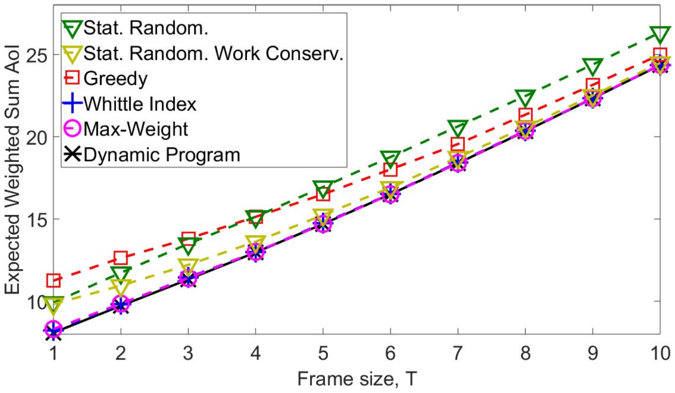

Fig. 6. Two-user general network with $K = 200, \alpha_1 = 2, \alpha_2 = 1, p_1 = 2/3, p_2 = 1/7$. The simulation result for each policy and for each value of T is an average over $2,000$ runs.

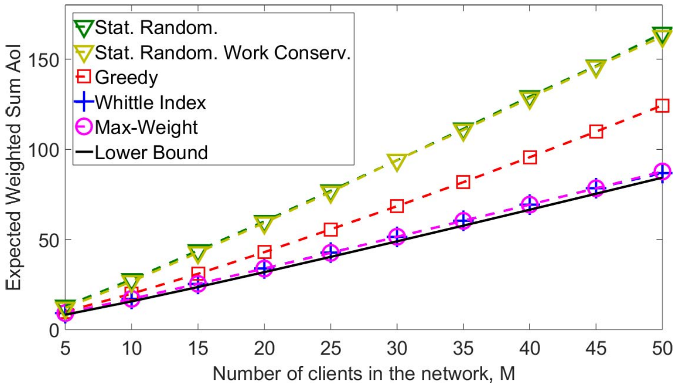

Fig. 7. Network with $T = 3, K = 50,000, \alpha_i = 1, p_i = i/M, \forall i$. The simulation result for each policy and for each value of M is an average over 10 runs.

Our results in Figs. 5 and 6 show the impact of work-conservation on the Randomized policy and show that the performances of the Max-Weight and Whittle Index policies are comparable to the optimal performance (DP) in every network setting considered. Moreover, the results in Fig. 5 support the optimality of the Greedy, Max-Weight and Whittle Index policies for any symmetric network. Figs. 6 and 7 suggest that, in general, the Max-Weight and Whittle Index Policies outperform Greedy and Randomized. An important feature of all policies examined in this paper is that they require low computational resources even for networks with a large number of clients.

VII. Concluding Remarks

This paper considered a wireless broadcast network with a BS sending time-sensitive information to multiple clients over unreliable channels. We studied the problem of optimizing scheduling decisions with respect to the expected weighted sum AoI of the clients in the network. Our main contributions include developing the Greedy, Randomized, Max-Weight and Whittle's Index policies; showing that for the case of symmetric networks, Greedy, Max-Weight and Whittle's Index are AoI-optimal; and deriving performance guarantees for all four low-complexity policies. Numerical results demonstrate the strong performances of the Max-Weight and Whittle's Index policies in a variety of network conditions.

The mathematical model in Sec. II describes a network that periodically generates packets at the BS and then transmits those packets to the clients. It is easy to see that the same model can represent other types of networks. A simple example is a polling network in which the BS requests packets from the clients and each client, once polled, generates fresh data and transmits that data back to the BS. This network with uplink traffic and on-demand generation of data can be represented by our model for the case $T = 1$. Interesting extensions of this work include considering stochastic arrivals, time-varying channels and multi-hop networks.

Appendix
Proof of Lemma 2

Lemma 2 (Round Robin): Without loss of generality, reorder the client index i in descending order of $\vec{h}_1$, with client 1 having the highest $h_{1,i}$ and client M the lowest $h_{1,i}$. The Greedy policy **delivers** packets according to the index sequence $(1, 2, \cdots, M, 1, 2, \cdots)$ until the end of the time-horizon K, i.e. Greedy follows a Round Robin pattern.

Proof: Suppose that $p_i = 1$ for all clients, meaning that every transmission is a successful packet delivery. Consider the first frame $k = 1$ and assume that there are less clients in the network than slots in a frame, i.e. $M < T$. In this case, the Greedy Policy delivers a packet to client 1 in the first slot, client 2 in the second slot, and so on, until the Mth packet is delivered. At this point, there are no undelivered packets left, and Greedy idles until the end of the frame. In the next frame $k = 2$, new packets are generated at the BS and the value of $h_{k,i}$ is updated to 1 for all clients. Since Greedy breaks ties arbitrarily, we choose to select clients in the same order $(1, 2, \cdots, M)$ during frame $k = 2$ and during all subsequent frames. This client ordering characterizes a circular order. Thus, for the case $M < T$ and $p_i = 1$, the Greedy Policy delivers packets to clients following a Round Robin pattern.

Now, consider the case $M \geq T$ and $p_i = 1$. Let $m_1 \in \mathbb{N}$ and $m_2 \in \{0, 1, \cdots, T-1\}$ be the quotient and remainder of the division of M by T, namely $M = m_1 T + m_2$. For simplicity of exposition, let the client index i be *reordered in descending order of $h_{k,i}$ at the beginning of every frame k*. Then, within every frame k, the Greedy Policy schedules clients in the following order $(1, \cdots, T)$. The evolution of the Greedy Policy is described in detail next:

- In the first frame, the Greedy policy delivers packets to clients 1 through T in order.
- At the beginning of the second frame, new packets are generated at the BS and the value of $h_{k,i}$ is updated to 1 for clients $\{1, \cdots, T\}$ and incremented by 1 for the remaining clients. Then, the client index i is reordered such that vector $\vec{h}_2$ is in descending order. Reordering can be accomplished with a cyclic shift of T elements, in particular, clients $\{1, \cdots, T\}$ become $\{M - T + 1, \cdots, M\}$ and clients that did not receive packets in the first frame have their index subtracted by T. With these reordered indexes, during the second frame, the Greedy policy delivers packets to clients 1 through T in order.
- Similarly, at the beginning of the third frame, new packets are generated at the BS and the value of $h_{k,i}$ is updated to 1 for clients $\{1, \cdots, T\}$ and incremented by 1 for the remaining clients. The vector $\vec{h}_3$ is reordered by applying the same cyclic shift of T elements. Notice that the value of $h_{3,i}$ is $h_{3,i} = 1$ for the clients that received packets in the second frame and $h_{3,i} = 2$ for the clients that received packets in the first frame. During the third frame, Greedy delivers packets to clients 1 through T in order.
- This process is repeated until frame $k = m_1$. Then, at the beginning of frame $k = m_1 + 1$, the reordered vector of $h_{k,i}$ is

$$\vec{h}_{m_1+1} = \begin{bmatrix} h_{1,i} + m_1 \\ m_1 \\ m_1 - 1 \\ \vdots \\ 2 \\ 1 \end{bmatrix} \begin{matrix} m_2 \text{ elements} \\ T \text{ elements} \\ T \text{ elements} \\ \vdots \\ T \text{ elements} \\ T \text{ elements} \end{matrix} \tag{60}$$

Clients $\{1, \cdots, m_2\}$ are the only ones that did not receive a packet so far. During frame $k = m_1 + 1$, the Greedy Policy delivers packets to clients 1 through T in order, where, by definition, $T > m_2$.
- Therefore, at the beginning of frame $k = m_1 + 2$, all clients have received at least one packet and the reordered vector of $h_{k,i}$ is

$$\vec{h}_{m_1+2} = \begin{bmatrix} m_1 + 1 \\ m_1 \\ m_1 - 1 \\ \vdots \\ 2 \\ 1 \end{bmatrix} \begin{matrix} m_2 \text{ elements} \\ T \text{ elements} \\ T \text{ elements} \\ \vdots \\ T \text{ elements} \\ T \text{ elements} \end{matrix} \tag{61}$$

During frame $k = m_1 + 2$, the Greedy policy delivers packets to clients 1 through T in order.
- At the beginning of frame $k = m_1 + 3$, the reordered vector of $h_{k,i}$ is

$$\vec{h}_{m_1+3} = \begin{bmatrix} m_1 + 1 \\ m_1 \\ m_1 - 1 \\ \vdots \\ 2 \\ 1 \end{bmatrix} \begin{matrix} m_2 \text{ elements} \\ T \text{ elements} \\ T \text{ elements} \\ \vdots \\ T \text{ elements} \\ T \text{ elements} \end{matrix} \tag{62}$$

Observe that (62) and (61) are identical. Clearly, in all frames that follow, the same sequence of events occur: i) vector $\vec{h}_k$ is updated according to (1); ii) vector $\vec{h}_k$ is reordered using a circular shift of T elements, resulting in $\vec{h}_k$ identical to (61); and iii) the Greedy Policy delivers clients 1 through T in order.

The description above for both cases $M < T$ and $M \geq T$ shows that when channels are error-free, namely $p_i = 1$, and we iteratively apply cyclic shifts of T elements to the client indexes, the Greedy Policy delivers packets to clients 1 through T in order at every frame k. Equivalently, when no cyclic shift is applied, the Greedy Policy delivers packets to clients in circular order.

When channels are unreliable, the only difference in the analysis is that each packet transmission to client i fails with probability $p_i \in (0, 1], \forall i$. According to Remark 1, in the event of a transmission failure, Greedy continues to transmit to the same client. Thus, transmission failures do not affect the order in which packets are *delivered*. Hence, irrespective of the network setup, the Greedy Policy delivers packets following a Round Robin pattern until the end of the time-horizon. ■

References

[1] I. Kadota, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Minimizing the age of information in broadcast wireless networks," in *Proc. IEEE Allerton*, Sep. 2016, pp. 844–851.

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

[3] R. D. Yates and S. Kaul, "Real-time status updating: Multiple sources," in *Proc. IEEE ISIT*, Jul. 2012, pp. 2666–2670.

[4] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in *Proc. IEEE ISIT*, Jun. 2015, pp. 1681–1685.

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

[6] C. Kam, S. Kompella, G. D. Nguyen, and A. Ephremides, "Effect of message transmission path diversity on status age," *IEEE Trans. Inf. Theory*, vol. 62, no. 3, pp. 1360–1374, Mar. 2016.

[7] K. Chen and L. Huang, "Age-of-information in the presence of error," in *Proc. IEEE ISIT*, Jun. 2016, pp. 2579–2583.

[8] E. Najm and R. Nasser, "Age of information: The gamma awakening," in *Proc. IEEE ISIT*, Jun. 2016, pp. 2574–2578.

[9] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Age and value of information: Non-linear age case," in *Proc. IEEE ISIT*, Jun. 2017, pp. 326–330.

[10] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," in *Proc. IEEE ITA*, Feb. 2015, pp. 25–31.

[11] B. T. Bacinoglu and E. Uysal-Biyikoglu, "Scheduling status updates to minimize age of information with an energy harvesting sensor," in *Proc. IEEE ISIT*, Jun. 2017, pp. 1122–1126.

[12] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *Proc. IEEE ISIT*, Jun. 2015, pp. 3008–3012.

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

[14] Q. He, D. Yuan, and A. Ephremides, "Optimizing freshness of information: On minimum age link scheduling in wireless systems," in *Proc. IEEE WiOpt*, May 2016, pp. 1–8.

[15] Q. He, D. Yuan, and A. Ephremides, "On optimal link scheduling with min-max peak age of information in wireless systems," in *Proc. IEEE ICC*, May 2016, pp. 1–7.

[16] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger, "Age-optimal constrained cache updating," in *Proc. IEEE ISIT*, Jun. 2017, pp. 141–145.

[17] S. K. Kaul and R. D. Yates, "Status updates over unreliable multiaccess channels," in *Proc. IEEE ISIT*, Jun. 2017, pp. 331–335.

[18] C. Joo and A. Eryilmaz, "Wireless scheduling for information freshness and synchrony: Drift-based design and heavy-traffic analysis," in *Proc. IEEE WiOpt*, May 2017, pp. 1–8.

[19] Y.-P. Hsu, E. Modiano, and L. Duan, "Age of information: Design and analysis of optimal scheduling algorithms," in *Proc. IEEE ISIT*, Jun. 2017, pp. 561–565.

[20] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Optimizing data freshness, throughput, and delay in multi-server information-update systems," in *Proc. IEEE ISIT*, Jul. 2016, pp. 2569–2573.

[21] 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.

[22] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Proc. IEEE SECON*, Jun. 2011, pp. 350–358.

[23] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Controlling the age of information: Buffer size, deadline, and packet replacement," in *Proc. IEEE MILCOM*, Nov. 2016, pp. 301–306.

[24] C. Kam, S. Kompella, and A. Ephremides, "Experimental evaluation of the age of information via emulation," in *Proc. IEEE MILCOM*, Oct. 2015, pp. 1070–1075.

[25] A. Franco, E. Fitzgerald, B. Landfeldt, N. Pappas, and V. Angelakis, "LUPMAC: A cross-layer MAC technique to improve the age of information over dense WLANs," in *Proc. IEEE ICT*, May 2016, pp. 1–6.

[26] P. P. Bhattacharya and A. Ephremides, "Optimal scheduling with strict deadlines," *IEEE Trans. Autom. Control*, vol. 34, no. 7, pp. 721–728, Jul. 1989.

[27] I.-H. Hou, V. Borkar, and P. R. Kumar, "A theory of QoS for wireless," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 486–494.

[28] K. S. Kim, C.-P. Li, I. Kadota, and E. Modiano, "Optimal scheduling of real-time traffic in wireless networks with delayed feedback," in *Proc. IEEE Allerton Conf. Commun., Control Comput.*, Sep./Oct. 2015, pp. 1143–1149.

[29] V. Raghunathan, V. Borkar, M. Cao, and P. R. Kumar, "Index policies for real-time multicast scheduling for wireless broadcast systems," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 1570–1578.

[30] B. Li, R. Li, and A. Eryilmaz, "Throughput-optimal wireless scheduling with regulated inter-service times," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1542–1552, Oct. 2015.

[31] B. Li, R. Li, and A. Eryilmaz, "Wireless scheduling design for optimizing both service regularity and mean delay in heavy-traffic regimes," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1867–1880, Jun. 2016.

[32] R. Singh, X. Guo, and P. R. Kumar, "Index policies for optimal mean-variance trade-off of inter-delivery times in real-time sensor networks," in *Proc. IEEE INFOCOM*, Apr./May 2015, pp. 505–512.

[33] X. Guo, R. Singh, P. R. Kumar, and Z. Niu, "A high reliability asymptotic approach for packet inter-delivery time optimization in cyber-physical systems," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2015, pp. 197–206.

[34] R. Singh and A. Stolyar, "MaxWeight scheduling: 'Smoothness' of the service process," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[35] X. Zheng, Z. Cai, J. Li, and H. Gao, "Scheduling flows with multiple service frequency constraints," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 496–504, Apr. 2017.

[36] B. Li, A. Eryilmaz, and R. Srikant, "Emulating round-robin in wireless networks," in *Proc. ACM MobiHoc*, 2017, Art. no. 21.

[37] D. Stoyan, *Comparison Methods for Queues and Other Stochastic Models* (Wiley Series in Probability and Statistics). Hoboken, NJ, USA: Wiley, 1983.

[38] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1, 3rd ed. Belmont, MA, USA: Athena Scientific, 2005.

[39] R. G. Gallager, *Stochastic Processes: Theory for Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[40] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan & Claypool, 2010.

[41] K. S. Kim, C.-P. Li, and E. Modiano, "Scheduling multicast traffic with deadlines in wireless networks," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 2193–2201.

[42] P. Whittle, "Restless bandits: Activity allocation in a changing world," *J. Appl. Probab.*, vol. 25, pp. 287–298, Jan. 1988.

[43] P. Mansourifard, T. Javidi, and B. Krishnamachari, "Optimality of myopic policy for a class of monotone affine restless multi-armed bandits," in *Proc. IEEE CDC*, Dec. 2012, pp. 877–882.

[44] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5547–5567, Nov. 2010.

[45] R. R. Weber and G. Weiss, "On an index policy for restless bandits," *J. Appl. Probab.*, vol. 27, no. 3, pp. 637–648, Sep. 1990.

[46] J. Gittins, K. Glazebrook, and R. Weber, *Multi-Armed Bandit Allocation Indices*, 2nd ed. Hoboken, NJ, USA: Wiley, Mar. 2011.

[47] A. Ganti, E. Modiano, and J. N. Tsitsiklis, "Optimal transmission scheduling in symmetric communication models with intermittent connectivity," *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 998–1008, Mar. 2007.

[48] Y.-C. Li and C.-C. Yeh, "Some equivalent forms of Bernoulli's inequality: A survey," *Appl. Math.*, vol. 4, no. 7, pp. 1070–1093, Jul. 2013.

Igor Kadota received the B.S. degree in electronic engineering from the Technological Institute of Aeronautics (ITA), Brazil, in 2010, the S.M. degree in telecommunications from ITA in 2013, and the S.M. degree in communication networks from the Massachusetts Institute of Technology (MIT) in 2016, where he is currently working toward the Ph.D. degree at the Laboratory for Information and Decision Systems (LIDS).

His research is on modeling, analysis, and design of communication networks, with the emphasis on wireless networks and real-time traffic.

Mr. Kadota received the Best Paper Award at the IEEE INFOCOM 2018 for his work on Age of Information.

Abhishek Sinha received the M.E. degree in telecommunication engineering from the Indian Institute of Science, Bengaluru, and the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 2012 and 2010, respectively, and the Ph.D. degree from the Massachusetts Institute of Technology (MIT) in 2017.

He was with the Laboratory for Information and Decision Systems, MIT. He will be starting as an Assistant Professor with the Department of Electrical Engineering, IIT Madras, in Fall 2018. His areas of interests include network control, information theory, optimization, and applied probability.

Dr. Sinha is a recipient of several awards, including the Best Paper Award in INFOCOM 2018, the Best Paper Award in MobiHoc 2016, the Prof. Jnansaran Chatterjee Memorial Gold Medal and T.P. Saha Memorial Gold Centered Silver Medal from Jadavpur University, and the Jagadis Bose National Science Talent Search (JBNSTS) Scholarship, Kolkata, India.

Elif Uysal-Biyikoglu received the B.S. degree as valedictorian from the Middle East Technical University (METU), Ankara, Turkey, in 1997, the S.M. degree in EECS from the Massachusetts Institute of Technology (MIT) in 1999, and the Ph.D. degree in EE from Stanford University in 2003. From 2003 to 2005, she was with MIT as a Lecturer.

She is currently a Professor in Electrical Engineering with METU, where she has been a Faculty Member since 2006. From 2005 to 2006, she was an Assistant Professor with the Ohio State University, where she was later an Adjunct Professor. She held visiting positions at MIT and Ohio State from 2014 to 2016.

Dr. Uysal-Biyikoglu is a recipient of the MIT Vinton Hayes Fellowship, the Stanford Graduate Fellowship, the NSF Foundations on Communication Grant from 2006 to 2010, the Turkish National Science Foundation KARIYER Award for 2007 to 2010, the 2014 Science Academy Young Scientist Award (in Turkey), and the 2010 IBM Faculty Award. Her research interests are at the junction of communication and networking theories and has provided consulting to the industry in these areas. She is an Associate Editor of the IEEE TRANSACTIONS ON NETWORKING and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

Rahul Singh received the B.E. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 2009, the M.Sc. degree in electrical engineering from the University of Notre Dame, South Bend, IN, USA, in 2011, and the Ph.D. degree in electrical and computer engineering from the Department of Electrical and Computer Engineering Texas A&M University, College Station, TX, USA, in 2015.

In 2015, he joined the Laboratory for Information Decision Systems, Massachusetts Institute of Technology, as a Postdoctoral Associate, where he was involved in several stochastic and adversarial decision making problems that arise in network control. He was with Encoredtech as a Data Scientist on machine learning problems arising in time-series modeling. He is currently with Intel as a Deep Learning Engineer. His research interests include decentralized control of large-scale complex cyberphysical systems, operation of electricity markets with renewable energy, machine learning, deep learning, and scheduling of stochastic networks serving real time traffic.

Eytan Modiano (F'12) 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 was a Naval Research Laboratory Fellow from 1987 to 1992, a National Research Council Postdoctoral Fellow from 1992 to 1993, and a member of the Technical Staff at MIT Lincoln Laboratory from 1993 to 1999. He is currently a Professor with the Department of Aeronautics and Astronautics and also an Associate Director with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT). His research is on modeling, analysis, and design of communication networks and protocols.

Dr. Modiano is 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 for the IEEE Wiopt 2006, the IEEE Infocom 2007, the ACM MobiHoc 2007, and the DRCN 2015. He is a fellow of the IEEE and an Associate Fellow of the AIAA. He is the Editor-in-Chief of the IEEE/ACM TRANSACTIONS ON NETWORKING, and served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY and the IEEE/ACM TRANSACTIONS ON NETWORKING. He has served for the IEEE Fellows committee in 2014 and 2015, respectively.