

# Age-of-Information-based Scheduling in Multiuser Uplinks with Stochastic Arrivals: A POMDP Approach

Aoyu Gong\*, Tong Zhang<sup>†</sup>, He Chen<sup>†</sup>, and Yijin Zhang\*

\*School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

<sup>†</sup>Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China

Email: gongaoyu@gmail.com, bennyzhangtong@yahoo.com, he.chen@ie.cuhk.edu.hk, yijin.zhang@gmail.com

**Abstract**—In this paper, we consider a multiuser uplink status update system, where a monitor aims to timely collect randomly generated status updates from multiple end nodes through a shared wireless channel. We adopt the recently proposed metric, termed age of information (AoI), to quantify the information timeliness and freshness. Due to the random generation of the status updates at the end node side, the monitor only grasps a partial knowledge of the status update arrivals. Under such a practical scenario, we aim to address a fundamental multiuser scheduling problem: how to schedule the end nodes to minimize the network-wide AoI? To solve this problem, we formulate it as a partially observable Markov decision process (POMDP), and develop a dynamic programming (DP) algorithm to obtain the optimal scheduling policy. By noting that the optimal policy is computationally prohibitive, we further design a low-complexity myopic policy that only minimizes the one-step expected reward. Simulation results show that the performance of the myopic policy approaches that of the optimal policy and is superior to that of the baseline policy.

## I. INTRODUCTION

The information freshness has become an increasingly important performance metric in this era of the Internet of Things (IoT). Various IoT services, such as remote monitoring and control, require the underlying information to be delivered as timely as possible [1], [2]. To quantify the information timeliness and freshness, the age of information (AoI) metric, defined as the time elapsed since the generation time of the latest received status update at the monitor, has been investigated in [3]–[7]. Early work (e.g., [4]–[10]) on the AoI focused on single-user systems, while recent work (e.g., [11]–[17]) shifted to multi-user systems, such as broadcast systems and multiuser uplink systems, where the AoI not only depends on the single-user behaviors but also depends on the interactions among different end nodes.

In broadcast systems, scheduling problems of minimizing the network-wide AoI were studied in [11]–[14]. The authors in [11] considered the “generate-at-will” model for the status update, where the status update arrivals could be generated by end nodes once they were scheduled to transmit. Three low-complexity scheduling policies were developed and analyzed in [11], including a randomized policy, a max-weight policy

and a Whittle’s index policy. The authors in [12] extended the work in [11] by studied the nonorthogonal multiple access. Considering event-triggered measurements where the status update arrivals are stochastic, the authors in [13] derived a universal lower bound of scheduling policies. In [14], both “generate-at-will” and stochastic arrival models with no buffer at end nodes were investigated, and an Whittle’s index policy was proposed to achieve the performance close to a structural Markov decision process (MDP) algorithm.

In multiuser uplink systems, scheduling problems of minimizing the network-wide AoI is more challenging than that in broadcast systems, especially when the status update arrivals are stochastic. This is mainly because the monitor may not know whether new status updates arrive at end nodes. Most existing work assumed end nodes used extra feedback overhead to report their status update arrivals so that the monitor had a complete knowledge of their status update arrivals [15], [16]. Such feedback leads to considerable overhead and thus makes the corresponding scheduling policies hard to implement in practice.

To combat this weakness, we consider a multiuser uplink system with stochastic status update arrivals. We assume that there is no extra feedback overhead for end nodes to report their status update arrivals. Thus, the monitor can obtain the status update arrival knowledge of an end node only when it is scheduled to transmit and its transmission is successful. Such a practical assumption leads to a partial knowledge of status update arrivals at the monitor. In this context, we aim to minimize the expected weighted-sum AoI (EWSAoI) of all end nodes by designing multiuser scheduling policies. Note that the consideration of a partial knowledge of status update arrivals renders difficulties in solving the scheduling policies in the considered system.

The main contributions of this paper are summarized as follows. We formulate the considered scheduling problem as a partially observable Markov decision process (POMDP), of which the belief state characterizes the fully observable AoI and the partially observable status update arrivals of end nodes at the monitor. Built on this POMDP, we develop a dynamic programming (DP) algorithm to solve the optimal policy. To reduce the computational complexity, we further propose a low-complexity myopic policy that only minimizes the one-step expected reward. Simulation results show that the performance of the myopic policy is very close to that of the

The work of H. Chen is supported by the CUHK direct grant under the project code 4055126. The work of Y. Zhang is supported by the Fundamental Research Funds for the Central Universities of China under Grant 30920021127.

The first two authors contributed equally to this paper. Any technical problems of this paper should go to H. Chen.

optimal policy solved by the DP algorithm and is superior to that of the baseline policy which utilizes no knowledge of status update arrivals. To the best of our knowledge, this is the first work that designs an information-freshness-oriented multiuser scheduling policy under partial system information.

## II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first describe the system model, and then formulate the network-wide AoI minimization problem.

### A. System Model

As shown in Fig. 1, we consider a multiuser uplink system where  $K$  end nodes report their freshest (i.e. most recently arriving) status updates to a common monitor. The time axis is divided into time slots of equal duration. Let  $T$  denote the finite horizon of the discrete-time system. In each time slot, a new status update arrives at end node  $i \in \{1, 2, \dots, K\}$  with probability  $\lambda_i \in (0, 1]$ ,  $\forall i$ . The Bernoulli arrival process is i.i.d. over time, and independent across different end nodes. At the beginning of each time slot, the monitor schedules at most one end node to transmit its freshest status update. The transmission of end node  $i$  to the monitor has a successful probability  $p_i$  and an error probability  $(1 - p_i)$ .

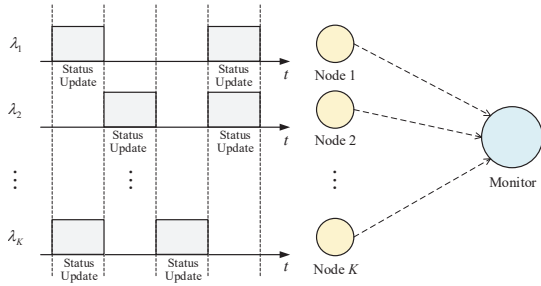


Fig. 1. The multiuser uplink system with stochastic arrivals of status updates.

### B. Local Age

We consider a single-packet buffer at each end node, i.e., the current status update at an end node is dropped when a new status update arrives at the end node. This assumption ensures status updates stored at each end node are freshest. We consider that no feedback is sent by the monitor after each transmission of status updates. Let the local age of end node  $i$ , denoted by  $z_i^t$ , represent the system time of the freshest status updates at end node  $i$ . The evolution of  $z_i^t$  can be expressed as follows:

$$z_i^{t+1} = \begin{cases} z_i^t + 1, & \text{no arrival in time slot } t, \\ 1, & \text{status update arrival in time slot } t. \end{cases} \quad (1)$$

As shown in (1), the local age of end node  $i$  will increase by 1 if there is no status update arrival and be reset to 1 otherwise. The local age of end nodes evolves independently according to the Bernoulli arrival processes. Without loss of generality, we assume that there is a status update for each end node at the beginning of each simulation runs.

In each time slot, reporting the local age of all end nodes to the monitor causes a large amount of extra overhead. For practical implementations, we enforce that the local age of an end node can be observed by the monitor only when the end node is scheduled and successfully transmits its freshest status update. This is because the status update received by the monitor contains the time-stamp of itself.

### C. AoI Minimization with Partial Knowledge of Arrivals

In this paper, we adopt the AoI metric to quantify the information freshness. The AoI of end node  $i$  at the monitor, denoted by  $h_i^t$ , will be set to the local age of end node  $i$ , if the end node is scheduled and its transmission is successful. Otherwise, the AoI of the end node will increase by 1. The evolution of  $h_i^t$  can be expressed as follows:

$$h_i^{t+1} = \begin{cases} z_i^t + 1, & \text{scheduled and received,} \\ h_i^t + 1, & \text{otherwise.} \end{cases} \quad (2)$$

In this context, the monitor is only aware of the local age of an end node that is scheduled and transmits successfully. This leads to partial observation of the system information at the monitor. With such partial knowledge, we aim to find a scheduling policy  $\pi$  minimizing the EWSAoI, which can be formulated as the following optimization problem:

$$(P1): \quad \min_{\pi} \frac{1}{TK} \mathbb{E} \left[ \sum_{t=1}^T \sum_{i=1}^K \omega_i h_i^t \mid \pi \right], \quad (3)$$

where  $\omega_i \in (0, \infty)$  is the importance weight of end node  $i$ . The expectation is taken over all system dynamics.

## III. POMDP FORMULATION

In this section, to solve the problem (P1), we reformulate it as a POMDP, and use the average reward of the POMDP to evaluate the EWSAoI.

- 1) **States:** We denote the state of end node  $i$  in time slot  $t$  by  $\mathbf{s}_i^t \triangleq [h_i^t, z_i^t]$ , where  $h_i^t \in \mathcal{T} \triangleq \{1, 2, 3, \dots\}$  is its instantaneous AoI at the monitor and  $z_i^t \in \mathcal{T}$  is its local age. Then, we denote the state of the POMDP in time slot  $t$  by  $\mathbf{s}^t \triangleq [\mathbf{h}^t, \mathbf{z}^t]$ , where  $\mathbf{h}^t \triangleq [h_1^t, h_2^t, \dots, h_K^t] \in \mathcal{H} \triangleq \mathcal{T}^K$  represents the AoI of all end nodes, and  $\mathbf{z}^t \triangleq [z_1^t, z_2^t, \dots, z_K^t] \in \mathcal{Z} \triangleq \mathcal{T}^K$  represents the local age of all end nodes. Denote by  $\mathcal{S}$  the space of all possible states.
- 2) **Actions:** We denote the action of the POMDP in time slot  $t$  by  $\mathbf{a}^t \triangleq [a_1^t, a_2^t, \dots, a_K^t]$ , where  $a_i^t \in \mathcal{A} \triangleq \{0, 1\}$  indicates whether end node  $i$  is scheduled to transmit or not in time slot  $t$ . If end node  $i$  is scheduled, then  $a_i^t = 1$ ; otherwise,  $a_i^t = 0$ . In the single-antenna system considered, the monitor can only schedule at most one end node in each time slot. Thus, we have  $\sum_{i=1}^K a_i^t \leq 1$ . Denote by  $\mathcal{A}$  the space of all possible actions.
- 3) **Observations:** The observations of the POMDP at the monitor consists of the fully observed AoI and partially observed local age of all end nodes. Specifically, if end node  $i$  is scheduled and its transmission is successful,

its local age can be accurately observed by the monitor. Otherwise, there is no observation of its local age. We denote the observation of the POMDP in time slot  $t$  by  $\mathbf{o}^t \triangleq [\mathbf{o}_1^t, \mathbf{o}_2^t, \dots, \mathbf{o}_K^t]$ , where  $\mathbf{o}_i^t \triangleq [h_i^t, \hat{z}_i^t]$  is the observation of end node  $i$ , including its fully observed AoI  $h_i^t$  and partially observed local age  $\hat{z}_i^t \in \{\mathcal{T}, X\}$ . Note that  $X$  means no observation of the local age of an end node, caused by its unsuccessful transmission or not being scheduled. Denote by  $\mathcal{O}$  the space of all possible observations.

- 4) Belief States: We denote the belief state of the POMDP in time slot  $t$  by  $\mathbf{I}^t \triangleq [\mathbf{h}^t, \mathbf{b}^t]$ , where  $\mathbf{b}^t$  is a probability distribution over  $\mathcal{Z}$ . Let  $b^t(\mathbf{z}^t)$  denote the probability assigned to state  $\mathbf{z}^t$  by distribution  $\mathbf{b}^t$ , which satisfies  $b^t(\mathbf{z}^t) \in [0, 1]$  for  $\forall \mathbf{z}^t \in \mathcal{Z}$  and  $\sum_{\mathbf{z}^t \in \mathcal{Z}} b^t(\mathbf{z}^t) = 1$ . It is worth mentioning that although in general the belief state of a POMDP is a probability distribution over  $\mathcal{S}$ , in our problem  $\mathbf{h}^t$  is fully observable, i.e., its belief state update is always deterministic given  $\mathbf{h}^{t-1}$ ,  $\mathbf{a}^{t-1}$  and  $\mathbf{o}^{t-1}$ . Denote by  $\mathcal{I}$  the space of all possible belief states.

- 5) State Transition Function and Observation Function: Because the belief state update of  $\mathbf{h}^t$  is deterministic, we only need to define the state transition function and observation function of  $\mathbf{z}^t$ . The state transition function is denoted by  $\Pr(\mathbf{z}^{t+1}|\mathbf{z}^t) = \prod_{i=1}^K \Pr(z_i^{t+1}|z_i^t)$ , giving the conditional probability of reaching state  $\mathbf{z}^{t+1}$  given state  $\mathbf{z}^t$ . For end node  $i$ , we have

$$\Pr(z_i^{t+1}|z_i^t) = \begin{cases} \lambda_i, & \text{if } z_i^{t+1} = 1, \\ 1 - \lambda_i, & \text{if } z_i^{t+1} = z_i^t + 1, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The observation function is denoted by  $\Pr(\mathbf{o}^t|\mathbf{z}^t, \mathbf{a}^t) = \prod_{i=1}^K \Pr(\mathbf{o}_i^t|z_i^t, a_i^t)$ , giving the conditional probability of making observation  $\mathbf{o}^t$  given state  $\mathbf{z}^t$  and action  $\mathbf{a}^t$ . If end node  $i$  is scheduled, then

$$\Pr(\mathbf{o}_i^t|z_i^t, 1) = \begin{cases} p_i, & \text{if } \hat{z}_i^t = z_i^t, \\ 1 - p_i, & \text{if } \hat{z}_i^t = X, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

If end node  $i$  is not scheduled, then  $\Pr(\mathbf{o}_i^t|z_i^t, 0) = 1$  if  $\hat{z}_i^t = X$ , and  $\Pr(\mathbf{o}_i^t|z_i^t, 0) = 0$  otherwise.

- 6) Belief State Update: In our POMDP, the monitor keeps belief states rather than knowing actual states. In time slot  $t$ , belief state  $\mathbf{I}^t$  is a sufficient statistic for a given history  $\{\mathbf{I}^1, \mathbf{a}^1, \mathbf{o}^1, \mathbf{a}^2, \mathbf{o}^2, \dots, \mathbf{a}^{t-1}, \mathbf{o}^{t-1}\}$ , consisting of two parts:  $\mathbf{h}^t$  and  $\mathbf{b}^t$ . When given  $\mathbf{I}^t$ ,  $\mathbf{a}^t$  and  $\mathbf{o}^t$ , for  $\forall i$ ,  $h_i^{t+1}$  can be updated as follows:

$$h_i^{t+1} = \begin{cases} \hat{z}_i^t + 1, & \text{if } \hat{z}_i^t \neq X, \\ h_i^t + 1, & \text{if } \hat{z}_i^t = X. \end{cases} \quad (6)$$

As shown in (6), the update of  $\mathbf{h}^t$  is always deterministic. When given the same condition, for  $\forall \mathbf{z}^{t+1} \in \mathcal{Z}$ ,

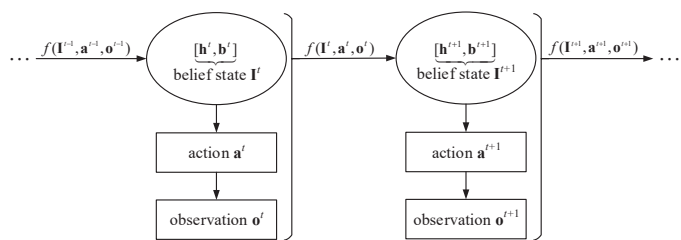


Fig. 2. An illustration of belief states, actions, observations and the update of belief states.

$b^{t+1}(\mathbf{z}^{t+1})$  can be updated via the Bayes' theorem:

$$b^{t+1}(\mathbf{z}^{t+1}) = \eta \sum_{\mathbf{z}^t \in \mathcal{Z}} \Pr(\mathbf{z}^{t+1}|\mathbf{z}^t) \Pr(\mathbf{o}^t|\mathbf{z}^t, \mathbf{a}^t) b^t(\mathbf{z}^t), \quad (7)$$

where

$$\eta = 1 / \sum_{\mathbf{z}^t \in \mathcal{Z}} \{\Pr(\mathbf{o}^t|\mathbf{z}^t, \mathbf{a}^t) b^t(\mathbf{z}^t)\}$$

is a normalizing factor. We denote the update of  $\mathbf{h}^t$  in (6) and the update of  $\mathbf{b}^t$  in (7) by the update function  $\mathbf{I}^{t+1} = f(\mathbf{I}^t, \mathbf{a}^t, \mathbf{o}^t)$ , of which the inputs are  $\mathbf{I}^t$ ,  $\mathbf{a}^t$  and  $\mathbf{o}^t$ , and the output is  $\mathbf{I}^{t+1}$ .

- 7) Reward: The expected immediate reward at belief state  $\mathbf{I}^t$  is defined as the weighted sum of the instantaneous AoI of all end nodes, i.e.,  $R(\mathbf{I}^t) \triangleq \sum_{i=1}^K \omega_i h_i^t$ . Then, the EWSAoI in (3) can be evaluated by

$$\frac{1}{TK} \mathbb{E} \left[ \sum_{t=1}^T R(\mathbf{I}^t) \middle| \mathbf{I}^1, \pi \right], \quad (8)$$

where  $\mathbf{I}^1$  is a given initial belief state.

- 8) Policy: In the above equation,  $\pi$  is a given policy defined as  $\pi = [\pi^1, \pi^2, \dots, \pi^T]$ , where  $\pi^t$  is a mapping from the belief space  $\mathcal{I}$  to the action space  $\mathcal{A}$ , i.e., decides which action  $\mathbf{a}^t$  should be taken when the POMDP is in belief state  $\mathbf{I}^t$ . Our aim is to find the optimal policy that minimizes the average reward in (8), i.e.,

$$\pi^* = \arg \min_{\pi} \frac{1}{TK} \mathbb{E} \left[ \sum_{t=1}^T R(\mathbf{I}^t) \middle| \mathbf{I}^1, \pi \right]. \quad (9)$$

To illustrate the POMDP formulation, we depict its belief states, actions, observations and update of belief states in Fig. 2.

#### IV. POLICY DESIGN FOR THE FORMULATED POMDP

In this section, we first propose a DP algorithm to find the optimal policy of the formulated POMDP and then devise a myopic policy with low computation complexity.

##### A. Dynamic Programming for the Optimal Policy

We follow [18] and resort to the DP framework for finding the optimal policy of the POMDP formulated in Section III. The DP method solves complex problems by breaking them down into a sequence of simpler sub-problems and then recursively combining solutions of sub-problems. It is worth

mentioning that the space of belief states of the POMDP is countable for any given initial belief state  $\mathbf{I}^1$ . We denote the finite set of belief states in time slot  $t$  by  $\mathcal{I}^t$ . The expected total reward of  $\forall \mathbf{I}^t \in \mathcal{I}^t$  can be denoted by the inner product  $\mathbf{b}^t \cdot \mathbf{V}(\mathbf{I}^t)$ , where  $\mathbf{V}(\mathbf{I}^t) \triangleq [v_{\mathbf{I}^t}(\mathbf{z}_1^t), v_{\mathbf{I}^t}(\mathbf{z}_2^t), \dots, v_{\mathbf{I}^t}(\mathbf{z}_{|\mathcal{Z}|}^t)]$  is a  $|\mathcal{Z}|$  dimensional vector by recalling that the update of  $\mathbf{h}^t$  is deterministic. Note that  $\mathbf{b}^t \cdot \mathbf{V}(\mathbf{I}^t)$  incorporates rewards from time slot  $t$  onward. The DP algorithm is formally described as follows.

---

**Algorithm 1** The DP Algorithm

---

- 1: Set  $\mathbf{V}^*(\mathbf{I}^T) = R(\mathbf{I}^T)\mathbf{1}_{|\mathcal{Z}|}$  for  $\forall \mathbf{I}^T \in \mathcal{I}^T$ , where  $\mathbf{1}_{|\mathcal{Z}|}$  is a  $|\mathcal{Z}|$  dimensional vector with all entries equal to 1, and  $t = T$ .
- 2: If  $t = 1$ , stop. Otherwise, set  $t = t - 1$  and go to step 3.
- 3: 1) For  $\forall \mathbf{I}^t \in \mathcal{I}^t$ , compute  $\pi^*(\mathbf{I}^t)$ .

$$\pi^*(\mathbf{I}^t) = \arg \min_{\mathbf{a}^t \in \mathcal{A}} \sum_{\mathbf{z}^t \in \mathcal{Z}} b^t(\mathbf{z}^t) \left[ R(\mathbf{I}^t) + \sum_{\mathbf{o}^t \in \mathcal{O}} \vartheta \times \Pr(\mathbf{o}^t | \mathbf{z}^t, \mathbf{a}^t) \sum_{\mathbf{z}^{t+1} \in \mathcal{Z}} \Pr(\mathbf{z}^{t+1} | \mathbf{z}^t) v_{\mathbf{I}^{t+1}}^*(\mathbf{z}^{t+1}) \right], \quad (10)$$

$\forall \mathbf{I}^{t+1} \in \mathcal{I}^{t+1}$ , where  $\vartheta = 1$  if  $f(\mathbf{I}^t, \mathbf{a}^t, \mathbf{o}^t) = \mathbf{I}^{t+1}$ , and  $\vartheta = 0$  otherwise.

- 2) For  $\forall \mathbf{I}^t \in \mathcal{I}^t$ , compute  $v_{\mathbf{I}^t}^*(\mathbf{z}^t)$  for  $\forall \mathbf{z}^t \in \mathcal{Z}$ .

$$v_{\mathbf{I}^t}^*(\mathbf{z}^t) = R(\mathbf{I}^t) + \sum_{\mathbf{o}^t \in \mathcal{O}} \vartheta \Pr(\mathbf{o}^t | \mathbf{z}^t, \pi^*(\mathbf{I}^t)) \times \sum_{\mathbf{z}^{t+1} \in \mathcal{Z}} \Pr(\mathbf{z}^{t+1} | \mathbf{z}^t) v_{\mathbf{I}^{t+1}}^*(\mathbf{z}^{t+1}), \quad \forall \mathbf{I}^{t+1} \in \mathcal{I}^{t+1}. \quad (11)$$

- 4: Return to step 2.
- 

The recursion simplifies the evaluation and optimization of  $\mathbf{V}^*(\mathbf{I}^1)$  over  $T$  time slots into a sequence of  $T - 1$  one-step computations. As shown in (11), in each step, the value of  $v_{\mathbf{I}^t}^*(\mathbf{z}^t)$  equals the immediate reward plus the expected total reward over the remaining time slots. The optimal policy  $\pi^*$  is defined as  $\pi^{*,t} : \mathbf{I}^t \rightarrow \pi^*(\mathbf{I}^t)$  for  $\forall \mathbf{I}^t \in \mathcal{I}^t$ . In particular,  $(\mathbf{b}^1 \cdot \mathbf{V}^*(\mathbf{I}^1)) / (TK)$  is the minimal EWSAoI given the initial belief state  $\mathbf{I}^1$ .

The DP algorithm represents an effective solution to find the optimal policy. However, the recursion is computationally prohibitive due to the following reasons. First, the AoI and local age tends to be large in real systems. Second, the dimension of the probability distribution  $\mathbf{b}^t$  grows exponentially with the number of end nodes. Thus, it is crucial to find a low-complexity policy.

### B. A Myopic Policy

In our problem, the local age of different end nodes evolves independently as described in Section II. As such, the monitor can only maintain probability distributions of the local age of each end node, which are sufficient statistics for the POMDP. We let  $\mathbb{B}^t = [\mathbf{b}_1^t, \mathbf{b}_2^t, \dots, \mathbf{b}_K^t]$  denote these distributions, where  $\mathbf{b}_i^t$  is the probability distribution of the local age of end node

$i$ . Let  $b_i^t(z_i^t)$  denote the probability assigned to local age  $z_i^t$  by distribution  $\mathbf{b}_i^t$ , satisfying  $b_i^t(z_i^t) \in [0, 1]$  for  $\forall z_i^t \in \mathcal{T}$ , and  $\sum_{z_i^t \in \mathcal{T}} b_i^t(z_i^t) = 1$ . Then, the belief state of the POMDP can be expressed as  $\mathbb{I}^t \triangleq [\mathbf{h}^t, \mathbb{B}^t]$ .

We then propose a myopic policy that minimizes the expected reward of the next time slot, also known as a one-step expected reward. Given  $\mathbb{I}^t$  for the POMDP, if action  $\mathbf{a}^t$  is chosen in time slot  $t$ , the one-step expected reward of the system is given by

$$\hat{R}(\mathbb{I}^t, \mathbf{a}^t) = \sum_{i=1}^K \omega_i \left[ (1 - a_i^t)(h_i^t + 1) + a_i^t \left( p_i \sum_{z_i^t \in \mathcal{T}} b_i^t(z_i^t)(z_i^t + 1) + (1 - p_i)(h_i^t + 1) \right) \right]. \quad (12)$$

For a scheduled end node, its expected AoI in next time slot is  $\sum_{z_i^t \in \mathcal{T}} b_i^t(z_i^t)(z_i^t + 1)$  with probability  $p_i$  or  $(h_i^t + 1)$  with probability  $(1 - p_i)$ . For an arbitrary unscheduled end node, its expected AoI in the next time slot is  $(h_i^t + 1)$ . The myopic policy for belief state  $\mathbb{I}^t$  can be obtained by

$$\tilde{\pi}^*(\mathbb{I}^t) = \arg \min_{\mathbf{a}^t \in \mathcal{A}} \hat{R}(\mathbb{I}^t, \mathbf{a}^t). \quad (13)$$

Then, the myopic policy  $\tilde{\pi}^*$  is defined as  $\tilde{\pi}^{*,t} : \mathbb{I}^t \rightarrow \tilde{\pi}^*(\mathbb{I}^t)$ . Compared with the optimal policy, the myopic policy is easier to implement. Not only  $\mathbb{B}^t$  reduces the dimension of  $\mathbf{b}^t$  from  $|\mathcal{T}|^K$  to  $K|\mathcal{T}|$ , growing linearly with the number of end nodes, but also the myopic policy only relies on the one-step expected reward instead of the expected total reward. The proposed myopic policy is formally described as follows.

---

**Algorithm 2** The Myopic Policy

---

- 1: Set  $t = 1$  and give initial belief state  $\mathbb{I}^1$ . For each end node, the monitor maintains its AoI  $h_i^t$  and a probability distribution  $\mathbf{b}_i^t$  of its local age.
- 2: In time slot  $t$ , the monitor chooses action  $\tilde{\mathbf{a}}^t$  by

$$\tilde{\mathbf{a}}^t = \arg \min_{\mathbf{a}^t \in \mathcal{A}} \hat{R}(\mathbb{I}^t, \mathbf{a}^t).$$

- 3: After taking action  $\tilde{\mathbf{a}}^t$ , the monitor makes observation  $\mathbf{o}^t$ . For each end node, the monitor updates its  $h_i^{t+1}$  by (6). If  $\hat{z}_i^t \neq X$ , the monitor updates its  $\mathbf{b}_i^{t+1}$  by

$$b_i^{t+1}(z_i^{t+1}) = \begin{cases} \lambda_i, & \text{if } z_i^{t+1} = 1, \\ 1 - \lambda_i, & \text{if } z_i^{t+1} = \hat{z}_i^t + 1, \\ 0, & \text{otherwise.} \end{cases}$$

Otherwise, the monitor updates its  $\mathbf{b}_i^{t+1}$  by

$$b_i^{t+1}(z_i^{t+1}) = \sum_{z_i^t \in \mathcal{T}} \Pr(z_i^{t+1} | z_i^t) b_i^t(z_i^t).$$

- 4: If  $t = T$ , stop. Otherwise, set  $t = t + 1$  and go to step 2.
-

## V. PERFORMANCE EVALUATION

In this section, after introducing a physical-layer model to obtain successful transmission probability  $p_i$ , we evaluate the DP algorithm and myopic policy via simulations.

### A. Successful Transmission Probability

In time slot  $t$ , the small-scale fading from end node  $i$  to the monitor is denoted by  $g_i^t$ , which is assumed to follow an exponential random variable with a unit mean. The large-scale fading is denoted by  $d_i^{-\tau}$ , where  $d_i$  is the distance from end node  $i$  to the monitor and  $\tau$  is the path-loss factor. The additive white Gaussian noise follows a complex Gaussian distribution  $\mathcal{CN}(0, \sigma^2)$ . The achievable rate is computed by  $r_i = \log_2(d_i^{-\tau} g_i^t P / \sigma^2 + 1)$ , where  $P$  is the transmit power, and the signal-to-noise ratio (SNR) is calculated by  $\text{SNR} = d_i^{-\tau} P / \sigma^2$ . If  $r_i$  is below the threshold  $r_{th}$ , the transmission of end node  $i$  is deemed to be unsuccessful. Consequently, the successful transmission probability of end node  $i$  can be obtained by

$$p_i = 1 - \Pr(r_i < r_{th}) = 1 - \Pr\left(g_i^t < \sigma^2 d_i^\tau \frac{2^{r_{th}} - 1}{P}\right),$$

$$\stackrel{(a)}{=} \exp\left(-\frac{\sigma^2 d_i^\tau (2^{r_{th}} - 1)}{P}\right),$$

where step (a) follows by the cumulative density function of the exponential random variable with a unit mean. We set the distance  $d_i = 5\text{m}$ , the path-loss factor  $\tau = 2$ , the threshold  $r_{th} = 1\text{bps/Hz}$  for all simulation runs in this section,  $\forall i, t$ .

### B. Comparisons with the Optimal Policy

In solving the POMDP-based policies, a state truncation  $D$  is applied to approximate the countable state space, i.e., the AoI and local age are both upper bounded by  $D$ ,  $\forall i, t$ .

Fig. 3 shows the analytical and simulation results of the optimal and myopic policies as a function of the SNR, in which we set  $K = 2$ ,  $T = 50$ ,  $D = 15$  and  $\omega_i = 1$ ,  $\forall i$ . Each simulation result is obtained from the average over  $10^6$  independent simulation runs. It is shown that, for both policies, the analytical and simulation results are well matched in all the simulated cases, which verifies the accuracy of the POMDP formulation. Moreover, the curves indicate that the performance of the myopic policy can be close to that of the optimal policy, i.e., the myopic policy can achieve near-optimal performance.

### C. Comparisons with Benchmarking Policies

We denote the proposed myopic policy by “**Myopic policy (partial knowledge), MPP**” and compare it with three benchmarking policies, described as follows:

- 1) **Myopic policy (full knowledge), MPF [17]**: assumes a full knowledge of status update arrivals at the monitor. Given  $\mathbf{s}^t$ , the monitor schedules end node  $i^*$  satisfying  $i^* = \arg \max_{\forall i} \omega_i p_i (h_i^t - z_i^t)$ .
- 2) **MaxAoI policy, MAP**: utilizes no knowledge of status update arrivals. Given  $\mathbf{h}^t$ , the monitor schedules end node  $i^*$  satisfying  $i^* = \arg \max_{\forall i} \omega_i p_i h_i^t$ .

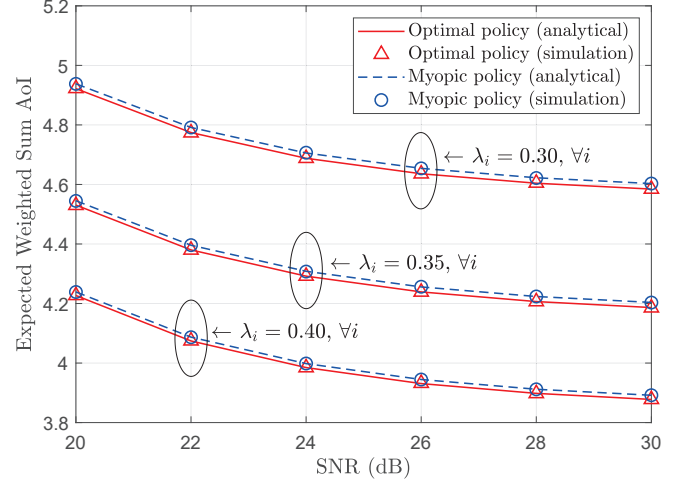


Fig. 3. The EWSAoI as a function of the SNR for different configurations.

- 3) **Randomized policy, RDP [13]**: schedules end nodes at random. Specifically, the monitor schedules end node  $i$  with probability  $\mu_i = \sqrt{\omega_i/p_i} / (\sum_{j=1}^K \sqrt{\omega_j/p_j})$ ,  $\forall i$ .

In this subsection, each simulation result is obtained from the average over 10 independent simulation runs.

Fig. 4 shows the simulation results of the four policies as a function of the number of end nodes, in which we set  $T = 10^6$ ,  $D = 50$ ,  $\text{SNR} = 25\text{dB}$ ,  $\lambda_i = 0.4$  and  $\omega_i = 1$ ,  $\forall i$ . We can see from Fig. 4 that the gap between MPP and MAP becomes smaller as the number of end nodes increases. This is because the gain from the partial knowledge of status update arrivals becomes trivial as the number of end nodes increases. Moreover, since MPF takes into account the full knowledge of the destination AoI and local age, it has the best performance among the four policies.

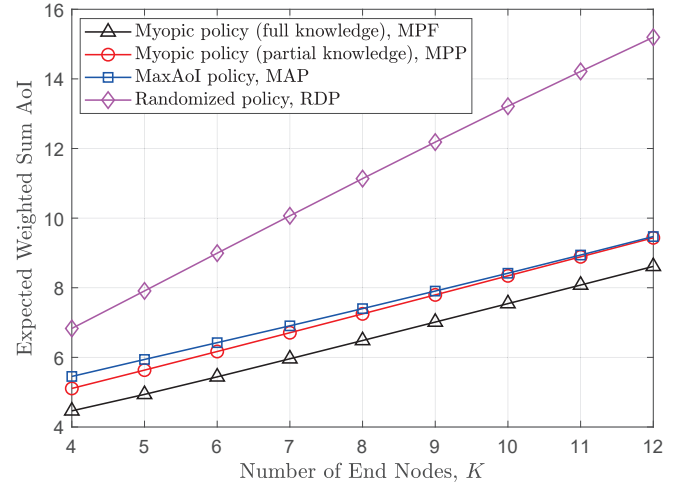


Fig. 4. The EWSAoI as a function of the number of end nodes.

Fig. 5 depicts the simulation results of the four policies as a function of the SNR, in which we set  $K = 5$ ,  $T = 10^6$ ,  $D = 50$ ,  $\lambda_i = 0.4$  and  $\omega_i = 1$ ,  $\forall i$ . The curves indicate



that the gap between MPP and MAP becomes larger as the SNR increases. The rationale is that, as the SNR increases, the monitor is more likely to accurately observe the local age of end nodes. In consequence, the partial system information at the monitor becomes closer to the full system information.

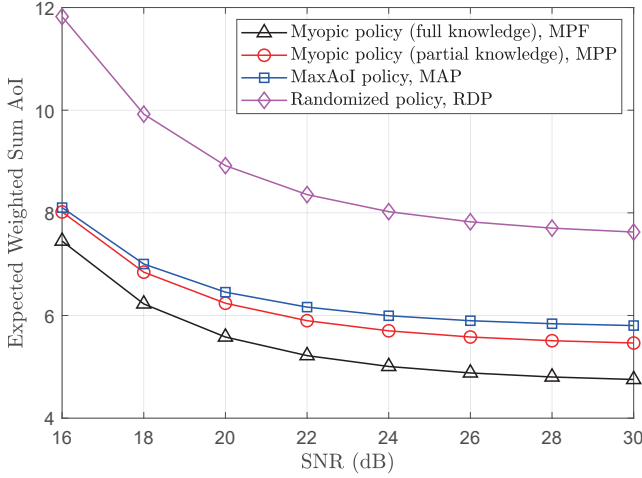


Fig. 5. The EWSAoI as a function of the SNR.

Fig. 6 shows the simulation results of the four policies as a function of the status update arrival rate, in which we set  $K = 5$ ,  $T = 10^6$ ,  $D = 50$ ,  $\text{SNR} = 25 \text{ dB}$  and  $\omega_i = 1, \forall i$ . We can observe that as the arrival rate increases, the performance of MPP, MPP and MAP turns to converge. This phenomenon is due to the fact that, for large arrival rates, the knowledge of status update arrivals is marginal. In particular, when  $\lambda_i = 1, \forall i$ , the considered arrival model will be equivalent to the “generate-at-will” arrival model, where there is no uncertainty about the status update arrivals.

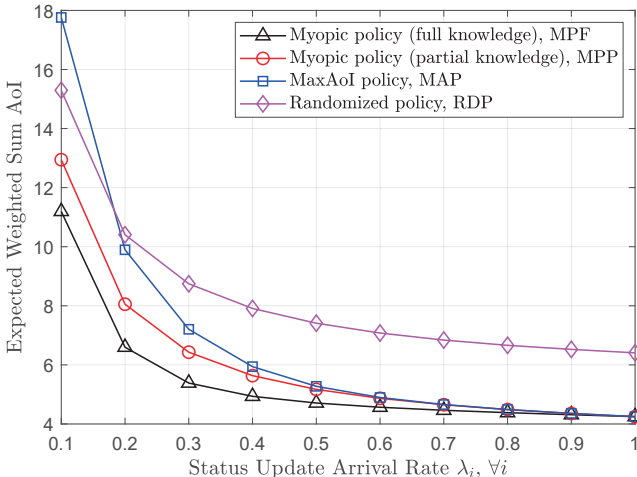


Fig. 6. The EWSAoI as a function of the status update arrival rate.

## VI. CONCLUSIONS

In this paper, we have investigated the information-freshness-oriented scheduling problem in the multiuser uplink

system, where the monitor has a partial knowledge of status update arrivals at the end node side. To tackle this problem, a partially observable Markov decision process has been formulated to characterize the dynamic behavior of such system. A dynamic programming algorithm has been developed to attain the optimal policy and a myopic policy with low computation complexity has further been devised. Simulation results have shown that the performance of the myopic policy approaches that of the optimal policy and is superior to that of the baseline policy.

## REFERENCES

- [1] A. Kosta, N. Pappas, and V. Angelakis, “Age of information: A new concept, metric, and tool,” *Found. Trends Netw.*, vol. 12, no. 3, pp. 162–259, 2017.
- [2] Y. Sun, I. Kadota, R. Talak, and E. Modiano, “Age of information: A new metric for information freshness,” *Synth. Lect. Commun.*, vol. 12, no. 2, pp. 1–224, 2019.
- [3] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, “Minimizing age of information in vehicular networks,” in *Proc. Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad-Hoc Commun. Netw. (SECOM)*, 2011, pp. 350–358.
- [4] S. Kaul, R. Yates, and M. Gruteser, “Real-time status: How often should one update?” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2012, pp. 2731–2735.
- [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, 2016.
- [6] C. Kam, S. Kompella, and A. Ephremides, “Age of information under random updates,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2013, pp. 66–70.
- [7] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, “Update or wait: How to keep your data fresh,” *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.
- [8] Y. Gu, H. Chen, Y. Zhou, Y. Li, and B. Vucetic, “Timely status update in internet of things monitoring systems: An age-energy tradeoff,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5324–5335, 2019.
- [9] Y. Gu, H. Chen, C. Zhai, Y. Li, and B. Vucetic, “Minimizing age of information in cognitive radio-based iot systems: Underlay or overlay?” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 273–10 288, 2019.
- [10] Q. Wang, H. Chen, Y. Gu, Y. Li, and B. Vucetic, “Minimizing the age of information of cognitive radio-based iot systems under a collision constraint,” *arXiv preprint arXiv:2001.02482*, 2020.
- [11] 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.
- [12] Q. Wang, H. Chen, Y. Li, and B. Vucetic, “Minimizing age of information via hybrid NOMA/OMA,” *arXiv preprint arXiv:2001.04042*, 2020.
- [13] I. Kadota and E. Modiano, “Minimizing the age of information in wireless networks with stochastic arrivals,” *IEEE Trans. Mobile Comput.*, pp. 1–1, 2019.
- [14] Y. Hsu, E. Modiano, and L. Duan, “Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals,” *IEEE Trans. Mobile Comput.*, pp. 1–1, 2019.
- [15] Z. Jiang, B. Krishnamachari, X. Zheng, S. Zhou, and Z. Niu, “Timely status update in wireless uplinks: Analytical solutions with asymptotic optimality,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3885–3898, 2019.
- [16] J. Sun, Z. Jiang, B. Krishnamachari, S. Zhou, and Z. Niu, “Closed-form whittles index-enabled random access for timely status update,” *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1538–1551, 2020.
- [17] H. Chen, Q. Wang, Z. Dong, and N. Zhang, “Multiuser scheduling for minimizing age of information in uplink MIMO systems,” *arXiv preprint arXiv:2002.00403*, 2020.
- [18] S. H. A. Ahmad, M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari, “Optimality of myopic sensing in multichannel opportunistic access,” *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4040–4050, 2009.