Sleep–Wake Sensor Scheduling for Minimizing AoI-Penalty in Industrial Internet of Things

Jia Wang, *Graduate Student Member, IEEE*, Xianghui Cao, *Senior Member, IEEE*, Bo Yin, *Student Member, IEEE*, and Yu Cheng, *Senior Member, IEEE*

***Abstract*—Ensuring data freshness is important for Industrial Internet of Things (IIoT). In this article, we consider a typical IIoT application where multiple sensors monitor some time-varying physical processes and report measurements to a central base station through an unreliable wireless channel. In order to save energy, each sensor may switch to sleep mode for a while after successfully transmitting a packet. Since only awake sensors are able to transmit data, we propose a novel function of Age of Information with penalty (or AoI-penalty) to capture the eagerness of the active sensors to provide fresh information. We formulate a new AoI-penalty minimization problem for scheduling the sensors' transmissions. We theoretically derive a necessary condition for the system's AoI-penalty to converge to a finite value, and further obtain a lower bound of the AoI-penalty. Moreover, we develop a max-weight-based scheduling policy and theoretically prove that it is the optimal policy when the network is symmetric and the channel is error-free. Simulation results demonstrate that the proposed policy achieves AoI performance near to the lower bound and that with such sleep–wake sensors, the achieved AoI performance is close to that with nonsleeping sensors but at a much lower energy cost.**

***Index Terms*—Age of Information (AoI), energy, lower bound, optimization, penalty, scheduling, sleep–wake sensors.**

I. INTRODUCTION

INDUSTRIAL Internet of Things (IIoT) refers to networks of sensors, controllers, actuators, and other devices to monitor and control industrial processes [1], [2]. As an emerging technology, IIoT has found a number of applications in smart factories, environmental monitoring, transportation, and so on [3]. IIoT systems usually run in real time where the freshness of data is of great importance [4], [5].

Consider a typical IIoT consisting of a number of sensors that monitor the dynamic physical process and report the status updates of the process to a base station (BS) for further data analysis, visualization, control, and/or decision making. The data freshness can be expressed by Age of Information (AoI), a performance metric that characterizes the time interval from the generation of the latest received information to the current time [6]–[11]. Enhancing the information freshness about the physical process encounters the challenging problem of minimizing the value of AoI. Due to mutual interference on the wireless channels, the sensors should not simultaneously transmit data to the BS, resulting in the problem of sensor scheduling for improving AoI performance. In conventional AoI minimization problems, sensors are often assumed to always remain active and thus can transmit data whenever scheduled [12], [13]. However, this may be unaffordable for commonly battery-powered sensors with limited energy in IIoT [14]. Sometimes it is unnecessary to sample the physical processes all the time when the process status changes slowly. Motivated by duty-cycling sensors that operate in sleep and active alternatively [15], [16], we introduce sleep modes for sensors to save energy.

This article considers a typical IIoT single-hop application scenario which has been described in the last paragraph. The sensors are programmed to work in a cyclic sleep–wake pattern. For each cycle, the sensor keeps in the sleep mode for a fixed sleep time first, and after that, it keeps active until it sends the updates successfully, then, it switches to the sleep mode again and a new cycle begins. The higher the emergency of the sensor, the shorter the sleep time is set. Moreover, the sensors in the sleep mode are allowed not to collect or send updates, so they are eager to provide fresh updates after they switch to the active mode, which naturally leads to that the AoI growth rate of active sensors should be higher than that of the sleeping ones. It also makes sense from the perspective of energy saving: the sensors want to send the data early so that they can switch to the sleep mode early to save energy. Hence, we propose a novel AoI-penalty function to capture the eagerness of information from active sensors. Specifically, the new AoI-penalty function grows at different rates, thus offering the system the property of overtaking which means that the AoI-penalty presently with a lower value has a chance to overtake that with a higher value afterward so that the corresponding sensor can have the opportunity to be scheduled.

On this basis, we investigate the sleep–wake sensor scheduling problem and formulate it as an AoI-penalty optimization problem. We propose a max-weight-based scheduling policy to solve the problem. At the beginning of every slot, the policy

Manuscript received May 10, 2021; revised August 11, 2021; accepted September 4, 2021. Date of publication September 13, 2021; date of current version April 25, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 92067111; in part by the Outstanding Young Scholarship of Jiangsu Province under Grant BK20180012; in part by the Defense Industrial Technology Development Program under Grant JCKY2020206B068; and in part by the Zhongying Young Scholars Project. The work of Yu Cheng was supported in part by NSF under Grant CNS-2008092 and Grant CNS-1816908.
(Corresponding author: Xianghui Cao.)

Jia Wang and Xianghui Cao are with the School of Automation, Southeast University, Nanjing 210096, China (e-mail: jwang97@seu.edu.cn; xhcao@seu.edu.cn).

Bo Yin and Yu Cheng are with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: byin@hawk.iit.edu; cheng@iit.edu).
Digital Object Identifier 10.1109/JIOT.2021.3112211

schedules an active sensor with the highest value of $\mathrm{MW}_i(k)$ (21) to send its updates. It is a lightweight policy because the time complexity is $\mathcal{O}(M)$ for every decision (M is the number of sensors in the network). Simulation results demonstrate that the AoI-penalty performance of the proposed policy is close to the theoretical lower bound, and that with the sleep–wake sensors, the AoI performance is close to that with nonsleep sensors but at a much lower energy cost. Besides, the results demonstrate that the AoI of the important sensors is lowered down. Our main contributions are summarized as follows.

1) We introduce a novel AoI-penalty function for sleep–wake sensors and formulate an AoI-penalty-minimal scheduling problem.
2) We analyze the convergence of the attainable AoI-penalty and develop a lower bound of it.
3) We prove that the max-weight scheduling policy is optimal if the network is symmetric and the channel is error-free, and that it will run in a Round Robin pattern when the network and the channel are symmetric.
4) We carry out simulations to evaluate the performance of the proposed policy.

The remainder of this article is organized as follows. Section II reviews more related work. Section III provides the system model, introduces the AoI-penalty function, and formulates the optimization problem. The convergence condition and lower bound of the problem solution are analyzed in Section IV. Section V introduces the max-weight policy and discusses its properties. Section VI demonstrates the simulation results, and finally Section VII concludes this article.

II. Related Work

Recently, as a metric of data freshness, AoI has attracted increasing attention for the growing demand of real-time data in communication networks. There are many variants of AoI research. For example, the AoI with priority has been studied. In [17], the updates from the sensors will send to the queue and the arrivals of the updates obey the Poisson distribution. They model the age process using a stochastic hybrid system for AoI. But they do not consider the energy consumption and the scheduling problem. Similar to [17], there are many studies that optimize AoI with the consideration of queuing theory. For example, to evaluate AoI for single-server queues [6] and queuing system with packet deadlines [18].

Sensor scheduling for minimizing AoI is an important research issue that has been studied in many works [12], [19]–[26]. Such a problem is often coupled with sampling for which the commonly considered sampling methods are arbitrary sampling, periodic sampling, and per time slot sampling [19]. For these, [19] develops a near-optimal scheduler that decides samples' transmissions to minimize AoI. In particular, with the per time slot sampling, sensors can collect status at the beginning of every slot [12], [20]–[22]. In [12], for the sensors under per time slot sampling, several scheduling methods, including optimal stationary randomized policy, max-weight policy, drift-plus-penalty policy, and Whittle's index policy, are derived and compared, and the results show that the max-weight and drift-plus-penalty policies outperform the others. In [20], an SQRT-Weight policy is proposed to schedule sensors to transmit their samples in networks with multiple sensors and multiorthogonal channels. For minimizing a nonlinear form of AoI, a weighted directed graph approach is adopted in [21] to decide sensor scheduling. There have been a few works considering other sampling methods [23], [24]. For example, in [23], considering that packets from different streams randomly arrive, both an optimal stationary randomized policy and a max-weight policy are developed to schedule the packets in three types of queues: 1) first-in–first-out queues; 2) single packets queues; and 3) no queues. In order to minimize the age of correlated information in which the information of application will not update until the latest status from the certain set of devices is received, a deep-reinforcement-learning-based approach is developed in [24] to schedule devices. However, the sensors in the above studies keep active all the time, which causes a waste of energy because many sensors in the active mode cannot be scheduled due to channel conflict. Thereby, we introduce the sleep mode for the sensor to save energy and study the sleep–wake sensor scheduling problem to minimize AoI.

In IIoT, devices, such as battery-powered sensors and energy-harvesting nodes, are energy constrained. Thus, how to achieve AoI minimization while saving energy is a challenging issue. In [27] where the sensors are assumed able to harvest energy with either random battery recharge or incremental recharge, the optimal status update policy to minimize AoI is obtained based on theories of renewal processes. In [28], four-power control schemes, including online scheduling, offline power control, save-and-transmit, and fixed power transmission, are considered to minimize both AoI and distortion. In [29], under transmission power constraints, the problem of multisensor scheduling is formulated and converted to a single-sensor-constrained Markov decision process, and then the scheduling policy is obtained by linear programming.

Aside from power control, sleep modes can be allowed for sensors in order to save their energy [30], [31]. In [30], when waking up from the sleep mode, a sensor first performs channel sensing and if it finds the channel busy, it goes back to sleep; otherwise, it transmits packets for status updating. To minimize AoI under the energy constraint, proper sleep time parameters are obtained. However, such a sleep model deprives the sensors of the opportunity to transmit their updates when finding a busy channel (because they will switch to the sleep mode afterward). Moreover, in some cases, it is hard for the sensors to sense whether the channel is busy or not. The channel sensing needs extra energy consumption. Therefore, in this article, it is the BS that schedules the sensors to send data, and the problem is formulated into a sleep–wake sensor scheduling model. The work in [31] studies the problem of minimizing the AoI of sleep–wake sensors in wireless sensor-actuator networks and develops a greedy-based scheduling policy. However, only the AoI of sensors in the active mode is accounted for. Moreover, compared with [31], we propose a novel AoI-penalty function to characterize the desire for fresh data from sleep–wake sensors in their active mode, while taking into account the sensors' AoI in both active

TABLE I
DEFINITION OF KEY NOTATIONS

Notation	Definition
M	Number of sensors in the network
K	Number of slots. Slot index is $k \in \{1, 2, \dots, K\}$
T_i	Sleep time of sensor i
$u_i(k)$	Indicator function shows whether sensor i is scheduled at the beginning of slot k
p_i	Probability of successful transmission from sensor i
$d_i(k)$	Indicator function shows whether the samples collected at he beginning of slot k by sensor i is received by the BS
$\mathbb{E}[\cdot]$	Expectation operator
$U_i(k)$	The generation time of the freshest data the BS receives from sensor i by the end of slot k
$j_i(k)$	The active time of sensor i at the beginning of slot k
$\Delta_i(k)$	The value of the AoI-penalty associated with sensor i at the beginning of slot k
w_i	The growth rate of AoI-penalty when sensor i is active
Δ_i^k	The area under the AoI-penalty $\Delta_i(k)$ curve for slot k
Δ^*	The optimal expected sum of AoI-penalty in (7)
L_B	The lower bound of the optimization problem (7)
$I_i[m]$	The number of slots between the $(m-1)$th and mth updates delivered by sensor i
$\overline{\mathbb{M}}[\mathbf{x}]$	The mean of a set of values $\mathbf{x}$

and sleep modes. The AoI-penalty takes energy consumption into account, which helps the system reduce energy consumption, and it helps the BS offer the important sensors more chances to send data.

III. System Model

Consider that M sensors monitor some physical processes and send measurements of the processes' states through an unreliable wireless channel to a BS. Ideally, all the sensors can send real-time measurements to the BS such that the processes' information at the BS is kept fresh all the time. However, practically there are some constraints on such an ideal model. First, simultaneous transmissions may collide with each other due to co-channel interference. In this article, we assume that at most one sensor can be scheduled at a time to transmit measurement data to the BS. Second, the wireless channel is lossy even though only one sensor transmits data. In addition, for energy-constrained sensors typical in IIoT, they are allowed to sleep to save energy. Specifically, once a data packet of sensor i is successfully received by the BS, the sensor is assumed to switch to sleep mode for a fixed period of sleep time T_i. The sleep time can be set in accordance with the physical processes and the importance of sensors, e.g., the sleep time of the sensor is set smaller when the physical process changes faster and the corresponding sensor's data is more important. Compared with the duty cycling sensors, the sleep–wake sensors in this article have more chances to provide fresh information. Because the sleep–wake sensors will sleep for a period of time only after the fresh data has been received by the BS. When the sensors are in sleep mode, the BS has the fresh data from the sensors, and it avoids the energy consumption that the sensors remain active when it cannot be scheduled due to channel conflict. However, the duty cycling sensors may still sleep when the data from them is stale because they cycle into sleep mode. The main notations used throughout this article are summarized in Table I.

Let the time be slotted with slot index $k \in \{1, 2, \dots, K\}$. For convenience, we set the length of each slot to 1 and assume that T_i is an integer for any i. We assume that the transmission delay for each sensor at any time is no more than one slot. At the beginning of each slot, the BS makes scheduling decisions, and at the end of each slot, the corresponding sensors which are scheduled will get deliveries of feedback to inform them whether their data is successfully received by the BS or not. Let $u_i(k) \in \{0, 1\}$ denote the choice of the BS at time k such that $u_i(k) = 1$ means sensor i is selected to transmit measurement data of its monitored process taken at the beginning of slot k, and $u_i(k) = 0$ means otherwise. As aforementioned, at most one sensor can be scheduled to transmit data, i.e., $\forall k$

$$\sum_{i=1}^{M} u_i(k) \leq 1. \tag{1}$$

We assume that a data transmission from sensor i will be successfully received by the BS with probability $p_i \in (0, 1]$ due to the lossy wireless channel. The channel is assumed to be stable and p_i does not change over time. Let $d_i(k) = 1$ denote that the BS successfully receives (by the end of slot k) the data from sensor i sent at that slot and $d_i(k) = 0$ denotes otherwise. Therefore

$$\mathbb{E}[d_i(k)] = p_i \mathbb{E}[u_i(k)]. \tag{2}$$

Denote by $U_i(k)$ the generation time of the freshest data the BS receives from sensor i by the end of slot k. Since the transmission delay is no more than 1, we have

$$U_i(k) = \begin{cases} U_i(k-1), & \text{if } d_i(k) = 0 \\ k, & \text{if } d_i(k) = 1. \end{cases} \tag{3}$$

A. Age of Information With Penalty

From the perspective of the BS, by conventional definition, the AoI associated with sensor i at the beginning of slot k is $k - U_i(k-1)$. Obviously, the AoI grows at a fixed rate (i.e., 1) when the BS does not receive fresher data from that sensor. AoI also can indicate the desire for fresh information. As a sleeping sensor is allowed not to transmit data, once it wakes up, it is eager to send its fresh data to the BS. If the sensor stays in active mode for a longer time, it not only means the sensor's information at the BS stales but more energy of the sensor is wasted for being active. Thus, it is reasonable that active sensors that have been in the active mode for a long time are more eager to provide fresh information than those who have just woken up. Therefore, we propose an *AoI-penalty* function to represent the property mentioned above, which makes the sensors that stay in the active mode for a long time easier to be scheduled [7], [32], [33]. Specifically, we set the growth rate of the AoI of sensor i as 1 and $w_i > 1$ when it is in sleep mode and active mode, respectively. For our system model, sensors are more eager to transmit information when they switch to the active mode. Because they can provide fresh data when they are active and they can switch to the sleep mode early to save energy if they send the data out early. The more urgent the sensor is (smaller T_i), the more eager it is to send data in the active mode, because smaller

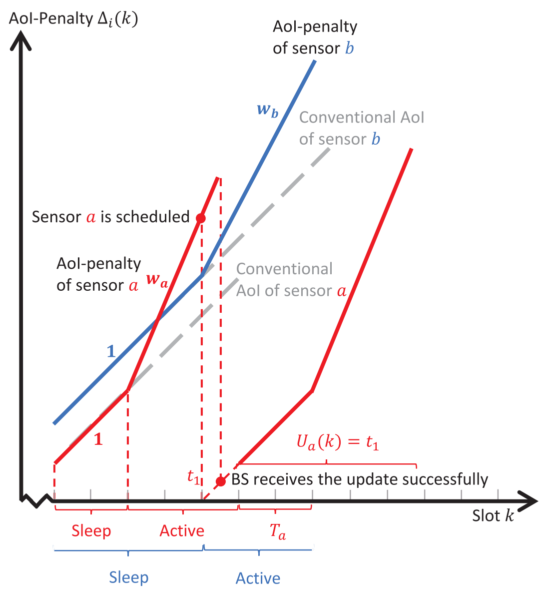


Fig. 1. Example of AoI-penalty.

T_i means the corresponding physical process is more important and changes faster. As a consequence, w_i and T_i should be negatively correlated and w_i should be greater than 1 but not too large. In addition, sensors with the same value of T_i have the same value of w_i. An example of the setting of w_i is provided in (25). In addition, the AoI-penalty function as introduced above brings new properties that traditional AoI does not have.

1) As shown in Fig. 1, consider that the BS schedules the sensor with the highest value of AoI at each time. With different growth rates, the AoI of sensor a is allowed to overtake that of sensor b, thus granting a the chance to be scheduled afterward. This overtaking property prevents the BS from scheduling the sensors that just wake up after a long sleep time rather than the sensors with short sleep time which have stayed in the active mode for a long time.
2) According to our definition, the sleep time can reflect the importance of sensors. The setting of AoI-penalty may offer the sensors with shorter sleep time (more important sensors) more chances to send their information due to the higher value of w_i.
3) The sensors in the active mode may be in sampling, transmitting data, or waiting, which consumes more energy than being in sleeping mode. In this sense, the proposed AoI-penalty function reflects the energy consumption of sleep–wake sensors. Optimizing AoI-penalty will also optimize energy consumption.

Upon successfully receiving the data from sensor i, the BS resets w_i as 1 for the time period from the data reception time to the end of the corresponding slot.

B. Optimization Problem

Let $j_i(k)$ indicate the number of slots that the sensor i has stayed in the active mode since it waked up from the last sleep mode, at the beginning of slot k. In particular, let $j_i(k) = -1$ if sensor i is in the sleep mode during slot k, then we have

$$j_i(k) = \begin{cases} k - Q_i(k-1), & \text{if } k \geq Q_i(k-1) \\ -1, & \text{otherwise} \end{cases} \tag{4}$$

where $Q_i(k) = U_i(k) + T_i + 1$.

Let Π be the set of all feasible scheduling policies. Denote by $\pi \in \Pi$ an arbitrary admissible policy. At the beginning of slot k, policy π be either keeping idle or selecting a sensor in the active mode to transmit update. Let $\Delta_i(k)$ represent the AoI-penalty of sensor i at the beginning of slot k

$$\Delta_i(k+1) = \begin{cases} 1, & \text{if } d_i(k) = 1 \\ \Delta_i(k) + 1, & \text{if } d_i(k) = 0 \text{ and } j_i(k) = -1 \\ \Delta_i(k) + w_i, & \text{if } d_i(k) = 0 \text{ and } j_i(k) \geq 0. \end{cases} \tag{5}$$

We adopt the expected sum of AoI-penalty to characterize the freshness of the information of the entire network when the BS employs policy π

$$\Delta^\pi = \frac{1}{KM}\mathbb{E}\left[\sum_{k=1}^{K}\sum_{i=1}^{M}\Delta_i(k)\middle|\vec{\Delta}(1), \vec{U}(0)\right] \tag{6}$$

where $\vec{\Delta}(1) = [\Delta_1(1)\ \Delta_2(1)\ \cdots\ \Delta_M(1)]^T$ and $\vec{U}(0) = [U_1(0)\ U_2(0)\ \cdots\ U_M(0)]^T$ are the vectors which indicate the initial state of the network. We assume that $\Delta_i(1) = 1$ and $U_i(0) = 0\ \forall i$, and omit $\vec{\Delta}(1)$ and $\vec{U}(0)$ henceforth.

Then, we can build an AoI-penalty optimization-scheduling problem

$$\min_{\pi\in\Pi} \lim_{K\to+\infty} \frac{1}{KM}\mathbb{E}\left[\sum_{k=1}^{K}\sum_{i=1}^{M}\Delta_i(k)\right] \tag{7a}$$

$$\text{s.t.} \begin{cases} u_i(k)j_i(k) \geq 0 & \forall k \quad \forall i \\ u_i(k) = \{0, 1\} & \forall k \quad \forall i \\ \sum_{i=1}^{M} u_i(k) \leq 1 & \forall k. \end{cases} \tag{7b}$$

Let Δ^* denote the optimal expected sum of AoI-penalty. There are three constraints in (7): the first constraint ensures that the selected sensor is in the active mode, the second one means a scheduling choice, while the last one ensures that at most one sensor is scheduled in a slot.

IV. Problem Analysis

A. Convergence Condition

First, we analyze the convergence of Δ^π when $K \to \infty$ so that we can narrow the search range of the scheduling policies. A convergence analysis shows the condition the scheduling policy π needs to meet if this policy makes Δ^π converge to a finite value.

Theorem 1 (Necessary Condition for Convergence): If a scheduling policy π makes Δ^π converge to a finite value as $K \to +\infty$, it should satisfy the following condition: $\forall t' \in [1, +\infty)$ and $\forall i \in \{1, 2, \ldots, M\}$

$$\Pr\left(\sum_{k=t'}^{\infty} u_i(k) \geq 1\right) = 1. \tag{8}$$

Proof: The proof is shown in Appendix A. ■

Theorem 1 reveals that the system will not appear as follows: with a positive probability, starting from a certain time

Algorithm 1: Obtaining the Lower Bound

Input: input parameters M, T_i, p_i, w, ζ
if $T_i \equiv 0$ **then**
  $\overline{\mathbb{M}}[I_i] \leftarrow \frac{1}{p_i}, \forall i$
  if *the second part of (14b) is not satisfied* **then**
    $L_B \leftarrow \frac{w^0\left(\sum_{i=1}^{M}\sqrt{1/p_i}\right)^2}{2M} + \frac{2-w^0}{2}$
  else
    $L_B \leftarrow \sum_{i=1}^{M} f(\overline{\mathbb{M}}[I_i])/M$
  end

else
  $\lambda \leftarrow 0$;
  $\overline{\mathbb{M}}[I_i] \leftarrow \max\left\{T_i + \frac{1}{p_i}, \sqrt{\frac{(w_i-1)(T_i+T_i^2)}{w_i}}\right\}, \forall i$
  if *the second part of (14b) is not satisfied* **then**
    $u \leftarrow min\{w_1, w_2, \ldots, w_M\}/(2M)$, $l \leftarrow 0$
    repeat
      $\lambda \leftarrow (l+u)/2$
      $\overline{\mathbb{M}}[I_i] \leftarrow \max\left\{T_i + \frac{1}{p_i}, \sqrt{\frac{(w_i-1)(T_i+T_i^2)}{w_i - 2M\lambda}}\right\}, \forall i$
      if $\sum_{i=1}^{M} \overline{\mathbb{M}}[I_i] \geq \left(\sum_{i=1}^{M}\sqrt{\frac{1}{p_i}}\right)^2$ **then**
        $u \leftarrow \lambda$
      else
        $l \leftarrow \lambda$
      end
    until $u - l \leq \zeta$;
  end
  $L_B \leftarrow \sum_{i=1}^{M} f(\overline{\mathbb{M}}[I_i])/M$
end
return L_B

slot, at least one sensor keeps idle all the time. According to Theorem 1, the scheduling policies should satisfy the necessary condition. Beyond that, Theorem 1 helps us prove some theorems in the following part.

B. Lower Bound of the Performance

Assume that there is an admissible policy π that meets all the constraints in (7b) and ensures each sensor will be scheduled during a finite time interval (Theorem 1). Denote by Ω the sample space of the network and by $\omega \in \Omega$ the sample path associated with policy π for the time horizon of K slots. For this sample path, let $D_i(K) = \sum_{k=1}^{K} d_i(k)$ be the total updates received by the BS during the K slots, and $I_i[m]$ be the interdelivery time associated with the deliveries of the updates come from sensor i, i.e., $I_i[m]$ is the number of slots between the $(m-1)$th and mth updates delivered by sensor i. Let R_i be the number of remaining slots after the last delivery of update by sensor i. Then, we can denote time horizon K as follows:

$$K = \sum_{m=1}^{D_i(k)} I_i[m] + R_i \quad \forall i \in \{1, 2, \ldots, M\} \tag{9}$$

where $I_i[m] \geq T_i + 1$, because sensor i is able to be scheduled only when it is in the active mode. We use the operator $\overline{\mathbb{M}}[\mathbf{x}]$ to represent the mean of a set of values $\mathbf{x}$. With such a definition, let the mean of $I_i[m]$ and $I_i^2[m]$ be

$$\overline{\mathbb{M}}[I_i] = \lim_{K\to+\infty} \frac{1}{D_i(K)} \sum_{m=1}^{D_i(K)} I_i[m] \tag{10}$$

$$\overline{\mathbb{M}}[I_i^2] = \lim_{K\to+\infty} \frac{1}{D_i(K)} \sum_{m=1}^{D_i(K)} I_i^2[m]. \tag{11}$$

Below we obtain a lower bound of the optimal solution.

Lemma 1: The optimal AoI-penalty of (7) meets the following:

$$\Delta^* \geq \frac{1}{M}\sum_{i=1}^{M} f(\overline{\mathbb{M}}[I_i]) \tag{12}$$

where

$$f(\overline{\mathbb{M}}[I_i]) = \frac{w_i\overline{\mathbb{M}}[I_i]}{2} + \frac{(w_i-1)(T_i+T_i^2)}{2\overline{\mathbb{M}}[I_i]} + \frac{2T_i - 2w_iT_i + 2 - w_i}{2}. \tag{13}$$

Proof: The proof is supplied in Appendix B. ■

Lemma 2: A lower bound L_B of the optimization problem (7) can be obtained by solving

$$L_B = \min_{\overline{\mathbb{M}}[I_i]} \frac{1}{M}\sum_{i=1}^{M} f(\overline{\mathbb{M}}[I_i]) \tag{14a}$$

$$\text{s.t.} \begin{cases} \overline{\mathbb{M}}[I_i] \geq T_i + \frac{1}{p_i} & \forall i \\ \sum_{i=1}^{M} \overline{\mathbb{M}}[I_i] \geq \left(\sum_{i=1}^{M}\sqrt{\frac{1}{p_i}}\right)^2. \end{cases} \tag{14b}$$

Proof: The proof is offered in Appendix C. ■

Theorem 2 (Lower Bound): L_B can be obtained by solving problem (14) with Algorithm 1. The w^0 in Algorithm 1 is the value of w when the sleep time T is 0.

Proof: The proof is provided in Appendix D. ■

Notice that ζ in Algorithm 1 is the parameter which influences the accuracy of the solution, and the value of it should be small. Theoretically, we have obtained the lower bound of the optimization problem in (7), which can help us analyze the performance of scheduling policies.

V. MAX-WEIGHT POLICY

It is difficult to obtain the optimal low-complexity solution to (7) because of its real-time scheduling decisions, randomness, infinite states, infinite time horizon, and the property of nonconvex ($u_i(k) \in \{0, 1\}$ makes the feasible set nonconvex). In [12], [13], [29], and [31], the optimal low-complexity solutions are also difficult to obtain for similar problems. Max-weight policy and drift-plus-penalty policy have better performance in [12]. Therefore, in this section, we adopt the lightweight scheduling policy: max-weight policy. It minimizes the drift of a Lyapunov function of the system state during every slot and it meets the condition in Theorem 1. Compared with [12] and [13], this article takes into account the sleep mode and nonlinear AoI, which makes it challenging to analyze the system's performance.

In this section, we introduce the proposed policy (or the MW policy for short) based on the concept of max-weight which has been used to deal with AoI scheduling problem in [12] and [13]. Using concepts from Lyapunov optimization [34], the MW policy can be obtained by minimizing the drift of a Lyapunov function of the system state during every slot. The state of the system during every slot is

$$S_k = \{\Delta_i(k), j_i(k)\}_{i=1}^{M}. \tag{15}$$

For every slot, define the quadratic Lyapunov function as

$$L(S_k) = \frac{1}{2}\sum_{i=1}^{M} \Delta_i^2(k) \tag{16}$$

which indicates the AoI-penalty of the system during slot k. Obviously, we can calculate $L(S_k)$ by the system state during slot k. We define Lyapunov drift which describes the expected change in function $L(S_k)$ from one slot to the next slot

$$G(S_k) = \mathbb{E}\big[L(S_{k+1}) - L(S_k)\,|\,S_k\big]. \tag{17}$$

Consequently, if we want to lower down the value of system AoI-penalty in (6), we need to keep $L(S_k)$ small by reducing $G(S_k)$ for every slot. Substituting (16) into (17), we can get

$$G(S_k) = \frac{1}{2}\sum_{i=1}^{M} \mathbb{E}[(\Delta_i^2(k+1) - \Delta_i^2(k)\,|\,S_k]. \tag{18}$$

Sensor i can be scheduled at the beginning of slot k only when it is in the active mode, i.e., $j_i(k) \geq 0$ or $\Delta_i(k) \geq T_i + 1$. Therefore, we just discuss the case where sensors are active

$$\mathbb{E}\Big[\Delta_i^2(k+1)\,|\,S_k\Big] = (\Delta_i(k) + w_i)^2(1 - p_i\mathbb{E}[u_i(k)\,|\,S_k]) + p_i\mathbb{E}[u_i(k)\,|\,S_k]. \tag{19}$$

Manipulating (19) yields that

$$\mathbb{E}[\Delta_i^2(k+1) - \Delta_i^2(k)\,|\,S_k] = -p_i\mathbb{E}[u_i(k)\,|\,S_k][(\Delta_i(k) + w_i)^2 - 1] + w_i(2\Delta_i(k) + w_i). \tag{20}$$

In order to reduce $G(S_k)$ during every slot, we need to determine the value of $\{u_i(k)\}_{i=1}^{M}$ at the beginning of every slot. We should consider (18) and (20) to determine the value of $u_i(k)$. Notice that only the first term of (20) can be affected by the choice of $u_i(k)$ of active sensors. So, we extract the first term of (20), remove $u_i(k)$ and minus sign

$$\mathrm{MW}_i(k) = p_i[(\Delta_i(k) + w_i)^2 - 1]. \tag{21}$$

Remark 1: In order to minimize (18), the MW policy selects the active sensor with the highest $\mathrm{MW}_i(k)$ to transmit the update at the beginning of each slot k. In addition, the policy is set to select the one with the smallest serial number to break the tie if two or more sensors have the same value of $\mathrm{MW}_i(k)$.

An example of the detailed operations of the MW policy is illustrated in Fig. 2. Assume that there are two sensors in the network. The parameters are $T_1 = 3$, $T_2 = 4$, $p_1 > p_2$, and

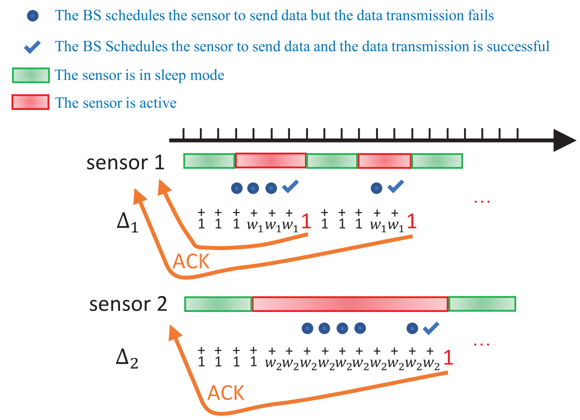


Fig. 2. Example of the operation of the MW policy.

$w_1 > w_2$. At the beginning, the two sensors are in sleep mode, and the BS cannot schedule any of them to send data. Then, sensor 1 and sensor 2 switch to active mode successively. The BS schedules sensor 1 to send data because the value of MW_1 is higher. When sensor 1 sends the data successfully, it will receive an acknowledgment from the BS, and then sensor 1 switches into the sleep mode (the BS knows sensor 1 will sleep for three slots). After that, the BS begins to schedule sensor 2 because it is still active. Assume that the transmission fails all the time and sensor 1 switches into the active mode again (the BS knows sensor 1 will be active at that time). Moreover, we assume that the value of MW_1 overtakes MW_2. Then, the BS stops scheduling sensor 2 and begins to schedule sensor 1. The later operation process can be seen in Fig. 2.

A. Performance Analysis

From (21), we can find that if one sensor is idle from a certain slot, there is at least one sensor's transmission fails all the time. However, p_i is greater than 0 for all i. The probability of such an event will not be a positive number. Therefore, we can obtain Remark 2.

Remark 2: The MW policy meets the necessary condition in Theorem 1.

Assume that the network is symmetric (all the sensors adopt the same sleep times $T_i \equiv T$ so that $w_i \equiv w$) and the channel is symmetric ($p_i \equiv p$). At the beginning of slot k, MW schedules sensor i because the value of $\Delta_i(k) + w$ is the highest among the active sensors. Assume this transmission fails and we can get $\Delta_i(k+1) = \Delta_i(k) + w$. The growth rates of AoI-penalty of other sensors are not greater than sensor i. According to (21) and $p_i \equiv p$, we have the following remark.

Remark 3: When the network and the channel are symmetric, once a sensor is scheduled, MW does not switch scheduling decisions until the sensor transmits the update successfully.

The following theorem reveals a special working pattern for MW when the network and the channel are symmetric.

Theorem 3 (Round Robin Pattern): When the network is symmetric, the necessary and sufficient condition of the MW policy generates the Round Robin pattern (the sensors transmit data in the order $1, 2, \ldots, M, 1, 2, \ldots, M, 1, 2, \ldots$) is that the channel is symmetric.

Proof: The proof is provided in Appendix E. ■

In fact, when the network and channel are symmetric, MW schedules one active sensor with the highest value of $\Delta_i(k)$ to transmit the update to the BS at the beginning of each slot k, which is the greedy policy in [13].

Remark 4: The MW policy will degenerate into the Greedy policy if the network and the channel are symmetric.

According to Theorem 3, we can get the following corollary.

Corollary 1: MW generates a Round Robin pattern when the network is symmetric and the channel is error-free ($p_i \equiv 1$).

Then, we can get some properties of MW when the network is symmetric and the channel is error-free.

Corollary 2: For a symmetric network and an error-free channel, as $K \to \infty$, the expected sum of AoI-penalty under MW becomes

$$\Delta^{\text{MW}} = \begin{cases} \frac{T}{2} + 1, \text{ if } M < T + 1 \\ f(M), \text{ if } M \geq T + 1 \end{cases} \tag{22}$$

where $f(\cdot)$ has been defined in (13).

Proof: According to Corollary 1, each sensor is in cycle mode after slot $T + M$. If $M < T + 1$ and $k \geq T + M + 1$, the change rule of the value of AoI-penalty for all the sensors during one transmission cycle is $\{1, 2, \ldots, T+1\}$. In addition, if $M \geq T + 1$, the change rule of the value of AoI-penalty for all the sensors during one transmission cycle is $\{1, 2, \ldots, T + 1, T + 1 + w, \ldots, T + 1 + (M - T - 1)w\}$. According to the renewal processes, we can get (22) by the strong law of large numbers [35]. ■

Next, we prove that MW is optimal when the network is symmetric and the channel is error-free.

Theorem 4 (Optimality of Max-Weight Policy): If the network is symmetric and the channel is error-free, as $K \to \infty$, the MW policy attains the minimum expected sum of AoI-penalty in (6) among all the admissible policies and it is the solution to the optimization problem in (7). Furthermore, the corresponding optimal AoI-penalty is

$$\Delta^* = \begin{cases} \frac{T}{2} + 1, \text{ if } M < T + 1 \\ f(M), \text{ if } M \geq T + 1. \end{cases} \tag{23}$$

Proof: The proof is provided in Appendix F. ■

In addition, according to the operation process in Section V, we can get Remark 5.

Remark 5: The MW policy needs to calculate the value of $\{\text{MW}_i(k)\}_{i=1}^M$ for every decision, so the time complexity of it is $\mathcal{O}(M)$.

B. Discussion

Based on the proposed MW policy, the scheduling decisions are made by the BS in a centralized manner at every slot. In order to make the decisions, the BS needs to know the current $\{\Delta_i\}_{i=1}^M$ and the working modes of the sensors. Fortunately, for each sensor i, by the definition of AoI in (5), $\Delta_i(k)$ is known by the BS, so is the working mode of sensor i. Therefore, during each slot, the BS can make the scheduling decision by himself, given the parameters $\{T_i\}_{i=1}^M$ and $\{p_i\}_{i=1}^M$. Specifically, the initial state of each sensor is assumed to be $\Delta_i(1) = 1$ and $U_i(0) = 0$. Given the BS the parameters $\{T_i\}_{i=1}^M$, it can calculate $\{w_i\}_{i=1}^M$ by the definition of the penalty function such as in (25). Each sensor will sleep for a fixed time T_i after successfully sending a data packet to the BS, and then keeps active until it is scheduled by the BS and successfully sends its current packet. Therefore, the BS also knows the working modes of the sensors during every slot.

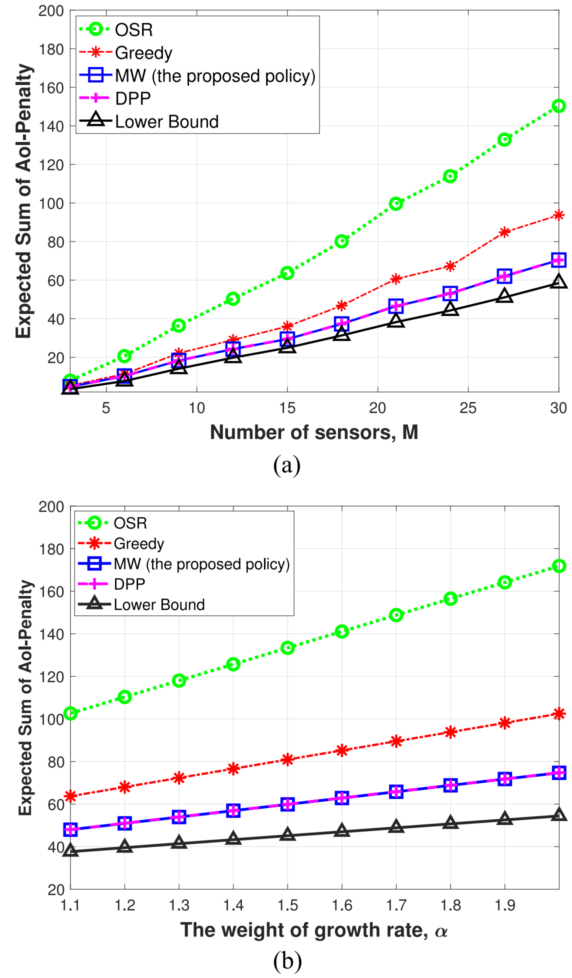


Fig. 3. Performance comparisons of different policies. (a) Comparison under different numbers of sensors M. (b) Comparison under different weight α.

In this article, we assume that the number of sensors in the system is a constant. However, our method can be easily extended to the cases with sensors join/leave the networks. Take that a new sensor joins the network for example. Upon finishing the necessary association processes with the BS, the sensor sends its sleep time, say T_i, to the BS. The BS evaluates the channel quality in terms of p_i. Then, the BS can make scheduling decisions based on our proposed method with the new sensor accounted.

VI. Simulation Results

In this section, we compare the performance of the proposed policy with three other scheduling policies: 1) greedy policy; 2) optimal stationary randomized policy (OSR); and 3) drift-plus-penalty policy (DPP). Different from [12], Stationary

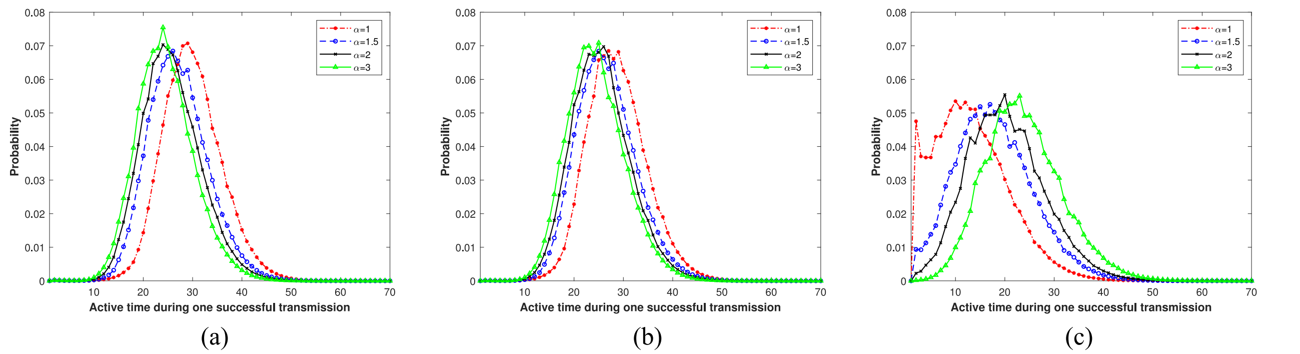


Fig. 4. Probability distribution of active time of the sensors with different emergency during one successful transmission. (a) Sleep time $T = 1$. (b) Sleep time $T = 4$. (c) Sleep time $T = 38$.

Randomized policy in this article schedules a sensor i with a fixed probability $\beta_i \in (0, 1]$ if the sensor is in active mode. According to the renewal process, we can derive a closed-form expression for (6) under Stationary Randomized Policy. Then, we can obtain the Optimal Stationary Randomized Policy OSR by using nonlinear optimization techniques. Given space limitations, we do not show how to solve the problem, but we will adopt OSR in simulations. The DPP policy adds a penalty term to the MW policy, and in our simulations, the penalty is set as

$$\frac{1}{2}\sum_{i=1}^{M}\ln\left(\frac{T_{\max}}{T_i}\right)\mathbb{E}[(\Delta_i(k+1)\,|\,S_k] \tag{24}$$

in the sense that sensors with short sleep time may gain more chances to be scheduled.

We evaluate the performance of these policies in terms of the expected sum of AoI-penalty as given in (6). Then, we use simulations to study the characteristics of our AoI-penalty function from the perspective of active time and the AoI of each sensor. Finally, we compare the performance in AoI and energy consumption between the sleep–wake sensors in this article and the nonsleeping sensors, such as in [12], [13], and [29]. The setting of w_i in the simulation is

$$w_i = \begin{cases} \alpha\left(1 + \frac{1-e^{-T_{\max}/T_i}}{1+e^{-T_{\max}/T_i}}\right), & \text{if } T_i \neq 0 \\ 2\alpha, & \text{if } T_i = 0 \end{cases} \tag{25}$$

where $T_{\max} = \max\{T_1, T_2, \ldots, T_M\}$. The weight $\alpha \geq 1$ represents how much the sensor is eager to send data in its active mode compared to the sleep mode. The above shows that w_i and T_i are negatively correlated and $1 \leq \alpha < w_i < 2\alpha$ ($T_i \neq 0$). The default parameter settings are $M = 20$, $K = 10^5$, $\alpha = 1$, and $p_i = i/M$, w_i is set as in (25). The sleep time T_i is randomly generated within $[1, 2M]$.

Fig. 3(a) shows the results of an average of 2000 simulations with fixed $K = 10^5$ and $\alpha = 1$, which shows simulation results with different $M \in \{3, 6, \ldots, 30\}$. Fig. 3(b) shows the results of an average of 2000 simulations with fixed $M = 20$ and $K = 10^5$, which shows simulation results with different $\alpha \in \{1.1, 1.2, 1.3, \ldots, 2.0\}$. From Fig. 3(a) and (b), we can find the MW policy and DPP policy have better AoI performance in our simulations than OSR policy and Greedy policy. The result of the MW policy and DPP policy is close to the lower bound.

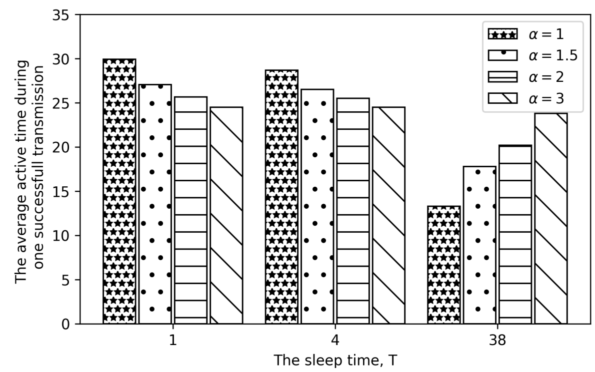


Fig. 5. Average active time with different emergency and different α during one successful transmission.

The performance of DPP and MW is similar because they only differ in the penalty term. Notice that the reason why OSR policy performs worst is that it wastes a lot of opportunities to schedule sensors.

Next, we do some simulations on the performance of the sensors of high and low emergency, respectively, where the differences in sensors' emergency are achieved by setting different sleep time periods. Fig. 4 demonstrates the distribution of the active time of three sensors of different emergency under the MW policy, where $p \equiv 0.5$, $K = 10^5$, and $M = 20$. Fig. 5 demonstrates the average active time of the sensors in Fig. 4. Fig. 4(a) and the left part in Fig. 5 represent the sensor with the highest emergency ($T = 1$), where we can see that the larger α is, the shorter the average active time of the sensor is. Most of the active time is between 20 and 50 slots. Fig. 4(b) and the middle part in Fig. 5 reflect the performance of the sensor with the second highest emergency ($T = 4$), and its performance is similar to that of the sensor with the highest emergency. In contrast, as shown in Fig. 4(c) and the right part in Fig. 5, the average active time of the sensor with the lowest emergency ($T = 38$) indicates that the larger α is, the larger the average active time of the sensor is. This is because the AoI-penalty function and the MW policy offer the important sensors more chances to send data but at the cost that the scheduling opportunities of unimportant sensors are sacrificed. The larger the value of α is, the more obvious the above property is.

In order to figure out whether our AoI-penalty function helps important sensors offer fresh information, we do further

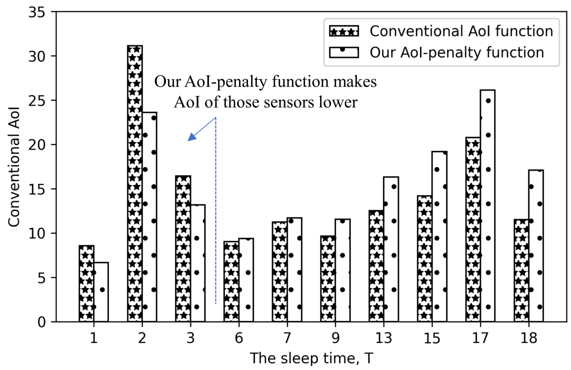


Fig. 6. Under different AoI model, the conventional AoI performance of each sensor.

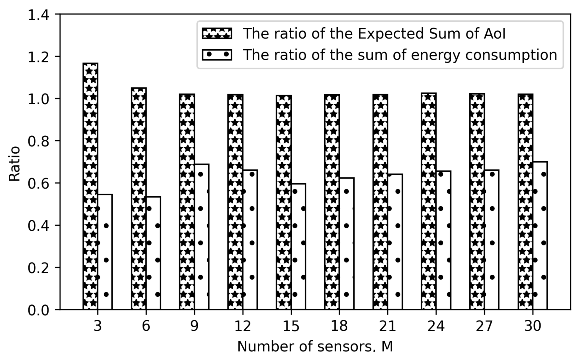


Fig. 7. Performance comparison of AoI and energy consumption between sleep–wake and nonsleeping sensors.

simulations. Fig. 6 shows the conventional AoI performance of each sensors, where $K = 10^5$, $M = 10$, and $\alpha = 3$ and the MW policy is adopted. In order to make AoI comparable, the AoI in Fig. 6 is conventional while our model operates in the AoI-penalty function. The conventional AoI grows at a fixed rate of 1 when no fresh information is provided [12]. The conventional AoI of the three sensors with shorter sleep time ($T = 1, 2, 3$) is lower in our AoI-penalty function, which demonstrates our AoI-penalty function sacrifices the AoI of the sensors with larger sleep time to provide smaller AoI for the sensors with smaller sleep time. Therefore, our AoI-penalty can provide fresh information for the important sensors (the sensors with smaller sleep time).

Then, in order to figure out whether sleep mode can help sensors strike a balance between fresh samples and energy conservation, we simulate two groups of sensors and compare the performance of AoI and energy consumption. The sensors in the first group are equipped with sleep mode while the sensors in the second group do not have sleep time and are nonsleeping. In order to make AoI comparable, the AoI in Fig. 7 is conventional while our model operates in sleep–wake and AoI-penalty function. Same to [36], we set the power consumption in sleeping, active, and transmitting data modes as 15 μW, 13.5 mW, and 24.75 mW, respectively. Fig. 7 shows the results of an average of 2000 simulations with $\alpha = 1$ and $K = 10^5$ under the MW policy. As shown, the AoI performance of the sensors equipped with sleep mode is quite close to the nonsleeping sensors while the sleep–wake sensors can save a lot of energy.

VII. Conclusion

In this article, the sensors can work in the sleep–wake mode to save energy and enhance lifetime. We design an AoI-penalty function to express the need for fresh information of active sensors and establish an AoI-penalty optimization-scheduling problem. Then, we introduce a max-weight-based scheduling policy and prove that it is the optimal policy when the network is symmetric and the channel is error-free. In addition, the MW policy will operate in a Round Robin pattern when the network and the channel are symmetric. In the simulation results, the AoI-penalty performance of the MW policy is close to the lower bound. It is also found that the penalty function lowers down the AoI of the important sensors. In addition, the simulation result shows that sleep–wake sensors in this article can save a lot of energy and provide a good AoI performance. In our future work, we will optimize AoI with a joint analysis of dynamic sleep time and scheduling policies when the sensors are equipped with limited energy.

Appendix A
Proof of Theorem 1

Under a scheduling policy π, let $\beta > 0$ be the probability of the following event: $\exists i', k' \in \mathbb{N}^+$, sensor i' is no longer scheduled after slot k'. When the event occurs as $K \to +\infty$, we assume the value of Δ^π still can converge to a finite value. Then, we adopt reduction to absurdity.

Without loss of generality, we can assume the last time the BS receives the update from sensor i' is the end of slot $k' - 1$. Then, we can get $U_{i'}(k) = U_{i'}(k'-1) = k'-1$ when $k \geq k'$. According to (6), we can get

$$\begin{aligned}\Delta^\pi &\geq \frac{\beta}{K}\sum_{k=1}^{K}\sum_{i=1}^{M}\Delta_i(k)\\ &= \frac{\beta}{K}\left(\sum_{k=1}^{K}\sum_{\substack{i=1\\ i\neq i'}}^{M}\Delta_i(k) + \sum_{k=1}^{k'-1}\Delta_{i'}(k) + \sum_{k=k'}^{K}\Delta_{i'}(k)\right). \quad (26)\end{aligned}$$

Extracting the last term of the above equation and applying the formula of the sum of arithmetic progression, we have

$$\begin{aligned}&\frac{\beta}{K}\sum_{k=k'}^{K}\Delta_{i'}(k)\\ &\quad= \frac{\beta[T_{i'}^2 + T_{i'}]}{2K} + \frac{\beta[j_{i'}(K)+1][2T_{i'}+2+w_{i'}j_{i'}(K)]}{2K}. \quad (27)\end{aligned}$$

The second term of (27) is infinite as $K \to \infty$ because the order of K on the molecule is higher. In addition, the first two terms of (26) are nonnegative and hence $\Delta^\pi \to \infty$ as $K \to \infty$, which conflicts with the hypothesis.

Appendix B
Proof of Lemma 1

Within each $I_i[m]$ interval, the initial value of $\Delta_i(k)$ is 1, the value of $\Delta_i(k)$ grows 1 every slot when sensor i is in the sleep mode and grows w_i every slot when sensor i is in the active mode. According to the limit we have mentioned, the value of R_i is finite even when $K \to +\infty$. Therefore, the sum of $\Delta_i(k)$ during the remaining time is finite and its influence on the value of (6) can be ignored when $K \to +\infty$. Then, the time-average AoI-penalty of sensor i when $K \to +\infty$ can be written as

$$\begin{aligned}
&\lim_{K\to+\infty} \frac{1}{K}\sum_{k=1}^{K} \Delta_i(k) \\
&= \lim_{K\to+\infty} \sum_{m=1}^{D_i(K)} \frac{T_i^2+T_i}{2K} \\
&\quad + \lim_{K\to+\infty} \sum_{m=1}^{D_i(K)} \frac{(I_i[m]-T_i)(2T_i+2+w_i(I_i[m]-T_i-1))}{2K} \\
&= \lim_{K\to+\infty} \frac{1}{K}\sum_{m=1}^{D_i(K)} \left(\frac{w_i}{2}I_i^2[m] + \frac{2T_i-2w_iT_i+2-w_i}{2} I_i[m]\right) \\
&\quad + \lim_{K\to+\infty} \frac{1}{K}\sum_{m=1}^{D_i(K)} \frac{(w_i-1)(T_i+T_i^2)}{2}.
\end{aligned} \tag{28}$$

Combining (9) and (10) yields

$$\lim_{K\to+\infty} \frac{K}{D_i(K)} = \overline{\mathbb{M}}[I_i] + \lim_{K\to+\infty} \frac{R_i}{D_i(K)}. \tag{29}$$

According to Theorem 1, the policy π we adopt makes $D_i(K) \to +\infty$ when $K \to +\infty$. Substituting (29) into (28) and applying the limit of R_i yield that

$$\begin{aligned}
&\lim_{K\to+\infty} \frac{1}{K}\sum_{k=1}^{K} \Delta_i(k) \\
&= \lim_{K\to+\infty} \frac{D_i(K)}{K}\frac{1}{D_i(K)} \sum_{m=1}^{D_i(K)} \frac{w_i}{2} I_i^2[m] \\
&\quad + \lim_{K\to+\infty} \frac{D_i(K)}{K}\frac{1}{D_i(K)} \sum_{m=1}^{D_i(K)} \frac{2T_i-2w_iT_i+2-w_i}{2} I_i[m] \\
&\quad + \lim_{K\to+\infty} \frac{D_i(K)}{K}\frac{1}{D_i(K)} \sum_{m=1}^{D_i(K)} \frac{(w_i-1)(T_i+T_i^2)}{2} \\
&= \frac{w_i\overline{\mathbb{M}}[I_i^2]}{2\overline{\mathbb{M}}[I_i]} + \frac{2T_i-2w_iT_i+2-w_i}{2} + \frac{(w_i-1)(T_i+T_i^2)}{2\overline{\mathbb{M}}[I_i]} \\
&\geq \frac{w_i\overline{\mathbb{M}}[I_i]}{2} + \frac{(w_i-1)(T_i+T_i^2)}{2\overline{\mathbb{M}}[I_i]} + \frac{2T_i-2w_iT_i+2-w_i}{2} \\
&\triangleq f(\overline{\mathbb{M}}[I_i])
\end{aligned} \tag{30}$$

where the above inequality applies Jensen's inequality. Taking the derivative of $f(\overline{\mathbb{M}}[I_i])$ with respect to $\overline{\mathbb{M}}[I_i]$, we can see that $f(\overline{\mathbb{M}}[I_i])$ is monotonically increasing when

$$\overline{\mathbb{M}}[I_i] \geq \sqrt{T_i^2+T_i-T_i^2/w_i-T_i/w_i}. \tag{31}$$

As mentioned above: $I_i[m] \geq T_i+1$, therefore

$$\overline{\mathbb{M}}[I_i] \geq T_i+1 > \sqrt{T_i^2+T_i-T_i^2/w_i-T_i/w_i} \tag{32}$$

which leads to that $f(\overline{\mathbb{M}}[I_i]) \geq f(T_i+1) > 0$. Since $\lim_{K\to+\infty}(1/K)\sum_{k=1}^{K}\Delta_i(k)$ is positive for all the admissible policies and all the sample path, by Fatou's lemma, we can obtain $\lim_{K\to+\infty}\mathbb{E}[(1/K)\sum_{k=1}^{K}\Delta_i(k)] \geq \mathbb{E}[\lim_{K\to+\infty}(1/K)\sum_{k=1}^{K}\Delta_i(k)]$.

As a consequence, from (7a), we can get $\Delta^* \geq (1/M)\sum_{i=1}^{M}\mathbb{E}[f(\overline{\mathbb{M}}[I_i])]$. Then, by removing the expectation operation due to the strong law of large numbers, inequality (12) is acquired.

Appendix C
Proof of Lemma 2

Assume that the policy π schedules sensor i once it switches to the active mode and does not schedule other active sensors unless the sensor i that is being scheduled has successfully transmitted its update to the BS. Then, the number of times sensor i is scheduled during a successful transmission interval is a geometric random variable. The behavior above leads to the lower bound of $\overline{\mathbb{M}}[I_i]$

$$\overline{\mathbb{M}}[I_i] \geq T_i + \frac{1}{p_i}. \tag{33}$$

Let $\Upsilon_i(K)$ represent the number of times the sensor i is scheduled up to and including slot K. Because the BS allows at most one sensor to be scheduled per slot, we can get

$$\sum_{i=1}^{M}\Upsilon_i(K) = \sum_{k=1}^{K}\sum_{i=1}^{M} u_i(k) \leq K. \tag{34}$$

With the strong law of large numbers, we know

$$\lim_{K\to+\infty}\frac{D_i(K)}{\Upsilon_i(K)} = p_i. \tag{35}$$

Combining (29), (34), and (35)

$$\begin{aligned}
\sum_{i=1}^{M}\overline{\mathbb{M}}[I_i] &= \lim_{k\to+\infty}\sum_{i=1}^{M}\frac{K}{D_i(K)} \\
&\geq \lim_{k\to+\infty}\left(\sum_{i=1}^{M}\Upsilon_i(K)\right)\left(\sum_{i=1}^{M}\frac{1}{D_i(K)}\right) \\
&\overset{(a)}{\geq} \lim_{k\to+\infty}\left(\sum_{i=1}^{M}\sqrt{\frac{\Upsilon_i(K)}{D_i(K)}}\right)^2 \\
&= \left(\sum_{i=1}^{M}\sqrt{\frac{1}{p_i}}\right)^2
\end{aligned} \tag{36}$$

where (a) adopts the Cauchy–Schwarz inequality. Combining the constrains with Lemmas 1 and 2 can be proved.

Appendix D
Proof of Theorem 2

Consider an arbitrary $i \in \{1,\ldots,M\}$. For the case: $T_i \equiv 0$, the second term in $f(\overline{\mathbb{M}}[I_i])$ is equal to zero and we can get

$$\frac{1}{M}\sum_{i=1}^{M} f(\overline{\mathbb{M}}[I_i]) = \sum_{i=1}^{M}\frac{w_i\overline{\mathbb{M}}[I_i]}{2M} + \sum_{i=1}^{M}\frac{2-w_i}{2M}. \tag{37}$$

The problem can be solved through linear programming. First, let $\overline{\mathbb{M}}[I_i] = 1/p_i \;\; \forall i$. If the second part of (14b) can be satisfied, we substitute $\overline{\mathbb{M}}[I_i] = 1/p_i$ into (14a) and get the value of L_B. However, if (14b) cannot be satisfied, we substitute the equation below into (14a) and get the value of L_B

$$\sum_{i=1}^{M} \overline{\mathbb{M}}[I_i] = \left(\sum_{i=1}^{M} \sqrt{\frac{1}{p_i}}\right)^2. \tag{38}$$

This process is realized from lines 1–7 in Algorithm 1.

For other cases, we can analyze the KKT conditions to get L_B. Let $\{\gamma_i\}_{i=1}^M$ be the KKT multipliers associated with the relaxation of the first part of (14b) and λ be the KKT multipliers associated with the relaxation of the second part of (14b). Then, for $\lambda \geq 0$ and $\gamma_i \geq 0$, we can define

$$\begin{aligned}\mathcal{L}(\overline{\mathbb{M}}[I_i], \lambda, \gamma_i) = &\frac{1}{M}\sum_{i=1}^{M} f(\overline{\mathbb{M}}[I_i]) \\ &+ \sum_{i=1}^{M} \gamma_i (T_i + \frac{1}{p_i} - \overline{\mathbb{M}}[I_i]) \\ &+ \lambda\left[\left(\sum_{i=1}^{M}\sqrt{\frac{1}{p_i}}\right)^2 - \sum_{i=1}^{M}\overline{\mathbb{M}}[I_i]\right]\end{aligned} \tag{39}$$

otherwise, we can define $\mathcal{L}(\overline{\mathbb{M}}[I_i], \lambda, \gamma_i) = +\infty$. So, we can obtain the KKT conditions.

1) The Stationarity is $\nabla_{\overline{\mathbb{M}}[I_i]}\mathcal{L}(\overline{\mathbb{M}}[I_i], \lambda, \gamma_i) = 0$.
2) The first Complementary Slackness is $\gamma_i(T_i + (1/p_i) - \overline{\mathbb{M}}[I_i]) = 0$.
3) The second Complementary Slackness is $\lambda[(\sum_{i=1}^{M}\sqrt{(1/p_i)})^2 - \sum_{i=1}^{M}\overline{\mathbb{M}}[I_i]] = 0$.
4) The Primal Feasibilities are $\overline{\mathbb{M}}[I_i] \geq T_i + (1/p_i)$ and $\sum_{i=1}^{M}\overline{\mathbb{M}}[I_i] \geq (\sum_{i=1}^{M}\sqrt{(1/p_i)})^2$.
5) The Dual Feasibilities are $\gamma_i \geq 0\ \forall i$ and $\lambda \geq 0$.

In order to get stationarity, $\nabla_{\overline{\mathbb{M}}[I_i]}\mathcal{L}(\overline{\mathbb{M}}[I_i], \lambda, \gamma_i) = 0$, we need to calculate the partial derivative of $\mathcal{L}(\overline{\mathbb{M}}[I_i], \lambda, \gamma_i)$ with respect to $\overline{\mathbb{M}}[I_i]$

$$\overline{\mathbb{M}}[I_i] = \sqrt{\frac{(w_i - 1)(T_i + T_i^2)}{w_i - 2M\gamma_i - 2M\lambda}}. \tag{40}$$

According to (40) and the KKT conditions, first, we assume that $\lambda = 0$, so the solution is

$$\overline{\mathbb{M}}[I_i] = \max\left\{T_i + \frac{1}{p_i}, \sqrt{\frac{(w_i - 1)(T_i + T_i^2)}{w_i}}\right\} \tag{41}$$

if the solution cannot meet the constraint of the second part of (14b), we can get $\lambda \neq 0$. If $\lambda \neq 0$, the solution is

$$\overline{\mathbb{M}}[I_i] = \max\left\{T_i + \frac{1}{p_i}, \sqrt{\frac{(w_i - 1)(T_i + T_i^2)}{w_i - 2M\lambda}}\right\} \tag{42}$$

where the solution should meet constraint 3) of KKT conditions. In this case, we need to find the solution of λ. Lines 8–24 in Algorithm 1 adopt the bisection method to get the solution of λ. Algorithm 1 can get the value of L_B by finding the optimal $\{\overline{\mathbb{M}}[I_i]\}_{i=1}^M$.

Appendix E
Proof of Theorem 3

First, we prove the necessary condition. Reorder the sensor index $\{i\}$ in descending order of p_i. At the beginning, all the sensors are in sleep mode. Then, during slot $T + 1$, all the sensors switch to the active mode and the MW policy selects the active sensor with the highest value of $\mathrm{MW}_i(T + 1)$ to transmit update. Manipulating (21), we have

$$\mathrm{MW}_i(k) = p_i\Delta_i^2(k) + 2p_i w\Delta_i(k) + p_i(w^2 - 1). \tag{43}$$

From Fig. 8, during slot $T + 1$, we can know sensor 1 is scheduled because the value of p_1 is highest among all the active sensors. Let $n_{i,j}$ represent the number of scheduling required for the jth successful transmission of sensor i. During slot $T+1+n_{1,1}$, the BS begins to schedule sensor 2. However, from slot $2T+1+n_{1,1}$ to $2T+1+n_{1,1}+n_{x,1}+\cdots+n_{M,1}-1$, sensor 1 is in the active mode again, which is shown in Fig. 8. Notice that x is the sensor that is being scheduled during slot $2T + 1 + n_{1,1}$. Although sensor x may have been scheduled several times before slot $2T + 1 + n_{1,1}$, it has no effect on the following derivation. From slot $2T+1+n_{1,1}$ to $2T+1+n_{1,1}+n_{x,1}+\cdots+n_{M,1}-1$, if sensor M has not finished its transmission task, we should keep $\mathrm{MW}_1(k) < \mathrm{MW}_M(k)$ during the time interval so that the scheduling order is $(1, 2, \ldots, M)$. As a consequence, for $z \in \{0, 1, 2, \ldots, n_{x,1}+\cdots+n_{M,1}-1\}$, we obtain the following inequality:

$$p_1\Big[(T + 1 + zw)^2 - 1\Big] < p_M\Big[\big(T + 1 + (T + n_{1,1} + z)w\big)^2 - 1\Big]. \tag{44}$$

Later, sensor 1 will be scheduled again. However, from slot $3T+1+n_{1,1}+n_{x,1}+\cdots+n_{M,1}+n_{1,2}$ to $3T+1+n_{1,1}+n_{x,1}+\cdots+n_{M,1}+n_{1,2}+n_{y,2}+\cdots+n_{M,2}-1$, sensor 1 is in the active mode again, which is shown in Fig. 8. The meaning of sensor y is like sensor x. Therefore, similar to the derivation in the previous paragraph, for $z \in \{0, 1, 2, \ldots, n_{y,2}+\cdots+n_{M,2}-1\}$, we obtain the following inequality:

$$p_1[(T + 1 + zw)^2 - 1] < p_M\Big[\big(T + 1 + (n_{1,2} + z)w\big)^2 - 1\Big]. \tag{45}$$

Then, the system goes into a cycle and the scheduling order is $(1, 2, \ldots, M, 1, 2, \ldots)$.

From (44), (45), $n_{1,1} \geq 1$, and $n_{1,2} \geq 1$, we can get

$$\frac{p_1}{p_M} < \frac{[T + 1 + (v + 1)w)]^2 - 1}{[T + 1 + vw)]^2 - 1} \tag{46}$$

where $v \geq 0$. We can deduce that the right-hand side of the inequality (46) decreases as v increases. We have

$$\frac{p_1}{p_M} < \lim_{v\to+\infty}\frac{[T + 1 + (v + 1)w)]^2 - 1}{[T + 1 + vw)]^2 - 1}. \tag{47}$$

Combining with $p_1 \geq p_M$, we can get

$$\frac{p_1}{p_M} = 1. \tag{48}$$

Moreover, p_1 and p_M are the highest and lowest value among all the p_i, respectively. As a consequence, we can get $p_i \equiv p$. Notice that we have ignored the situation that BS is

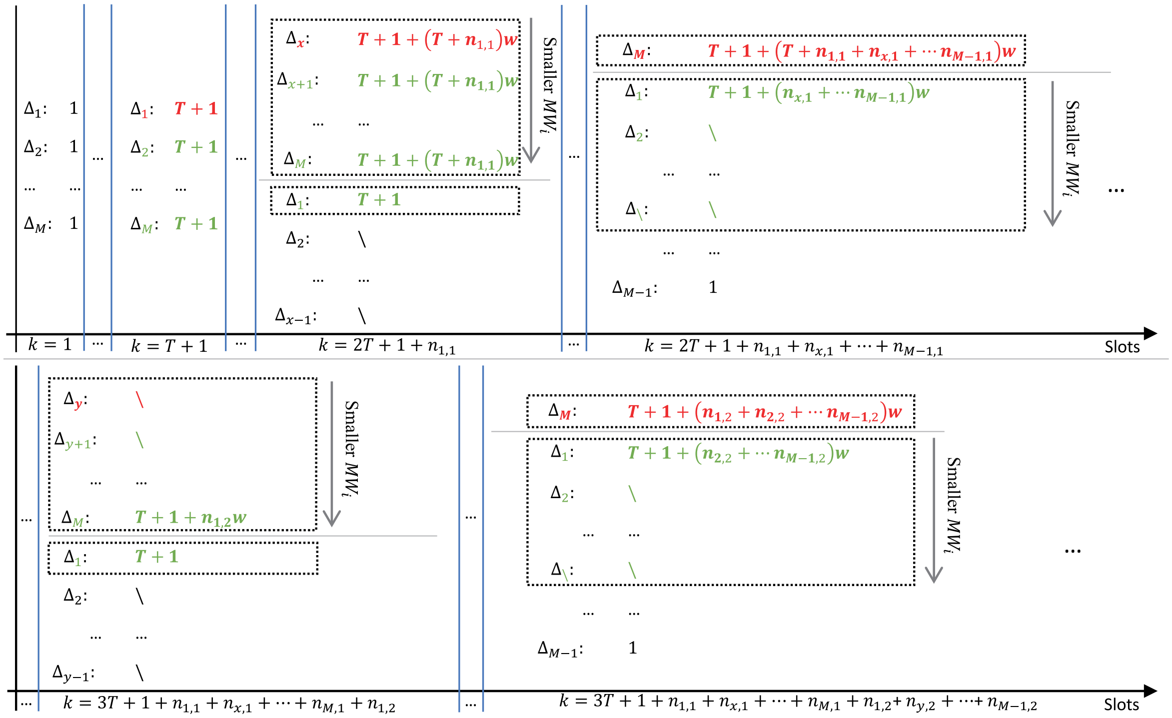


Fig. 8. Evolution of AoI-penalty under the MW policy, where the network is symmetric.

idle, because it is obvious that the scheduling order still is $(1, 2, \ldots, M, 1, 2, \ldots)$ in such a situation.

Then, we can prove the sufficient condition. Lower AoI-penalty will lead to lower value of (43) when $p_i \equiv p$. According to Remark 3, the scheduling order is $(1, 2, \ldots, M, 1, 2, \ldots)$ when the network and channel are symmetric.

Appendix F
Proof of Theorem 4

Denote by $\mathbb{I}_i(\cdot)$ the indicator function in the sense that $\mathbb{I}_i(s) = 1$ if $i \notin s$ and $\mathbb{I}_i(s) = 0$ otherwise. Let $\hat{s}_k$ be the sensor that is scheduled during slot k and transmits the update successfully, and let $\hat{c}_k$ be the set of sensors that is in the sleep mode during slot k. Let $\vec{\mathbb{I}}(\cdot) = [\mathbb{I}_1(\cdot)\ \mathbb{I}_2(\cdot)\ \cdots\ \mathbb{I}_M(\cdot)]^T$ and let $\vec{\Delta}(k) \odot \vec{\mathbb{I}}(\cdot)$ be the entrywise product of vectors $\vec{\Delta}(k)$ and $\vec{\mathbb{I}}(\cdot)$. Then, $\vec{\Delta}(k)$ evolves as

$$\vec{\Delta}(k+1) = \vec{\Delta}(k) \odot \vec{\mathbb{I}}(\hat{s}_k) + (w-1)\vec{\mathbb{I}}(\hat{s}_k \cup \hat{c}_k) + \vec{1} \quad (49)$$

where $\vec{1}$ is the unity column vector with length M.

Then, we can get that $\vec{\Delta}(k+1)$ can be expressed as a function of $\vec{\Delta}(1)$, $\{\hat{s}_j\}_{j=1}^k$, and $\{\hat{c}_j\}_{j=1}^k$

$$\vec{\Delta}(k+1) = \vec{\Delta}(1) \odot \vec{\mathbb{I}}\left(\bigcup_{j=1}^{k} \hat{s}_j\right) + \sum_{a=1}^{k}(w-1)\vec{\mathbb{I}}\left(\bigcup_{j=a}^{k} \hat{s}_j \cup \hat{c}_a\right) + \sum_{a=2}^{k}\vec{\mathbb{I}}\left(\bigcup_{j=a}^{k} \hat{s}_j\right) + \vec{1}. \quad (50)$$

We prove (50) by induction. Substituting $k = 1$ into (50) and we can get (49). Then, assume that (50) holds for k. For step $k+1$, substituting $\vec{\Delta}(k)$ of the form (50) into (49) yields that

$$\begin{aligned}\vec{\Delta}(k+1) &= \vec{\Delta}(k) \odot \vec{\mathbb{I}}(\hat{s}_k) + (w-1)\vec{\mathbb{I}}(\hat{s}_k \cup \hat{c}_k) + \vec{1}\\ &= \left[\vec{\Delta}(1) \odot \vec{\mathbb{I}}\left(\bigcup_{j=1}^{k-1} \hat{s}_j\right) + \sum_{a=1}^{k-1}(w-1)\vec{\mathbb{I}}\left(\bigcup_{j=a}^{k-1} \hat{s}_j \cup \hat{c}_a\right) + \sum_{a=2}^{k-1}\vec{\mathbb{I}}\left(\bigcup_{j=a}^{k-1} \hat{s}_j\right) + \vec{1}\right] \odot \vec{\mathbb{I}}(\hat{s}_k)\\ &\quad + (w-1)\vec{\mathbb{I}}(\hat{s}_k \cup \hat{c}_k) + \vec{1}\\ &= \vec{\Delta}(1) \odot \vec{\mathbb{I}}\left(\bigcup_{j=1}^{k} \hat{s}_j\right) + \sum_{a=1}^{k}(w-1)\vec{\mathbb{I}}\left(\bigcup_{j=a}^{k} \hat{s}_j \cup \hat{c}_a\right)\\ &\quad + \sum_{a=2}^{k}\vec{\mathbb{I}}\left(\bigcup_{j=a}^{k} \hat{s}_j\right) + \vec{1} \end{aligned} \quad (51)$$

which is identical to the expression in (50). By induction, we know that (50) holds for all $k \geq 1$.

Then, we can get

$$\begin{aligned}\sum_{i=1}^{M}\Delta_i(k) &= \sum_{i=1}^{M}\left[\Delta_i(1)\mathbb{I}_i\left(\bigcup_{j=1}^{k-1} \hat{s}_j\right) + \sum_{a=1}^{k-1}(w-1)\mathbb{I}_i\left(\bigcup_{j=a}^{k-1} \hat{s}_j \cup \hat{c}_a\right)\right.\\ &\quad \left. + \sum_{a=2}^{k-1}\mathbb{I}_i\left(\bigcup_{j=a}^{k-1} \hat{s}_j\right) + 1\right]\end{aligned}$$

$$= \sum_{i=1}^{M} \Delta_i(1)\mathbb{I}_i\left(\bigcup_{j=1}^{k-1}\hat{s}_j\right) + \sum_{i=1}^{M}\sum_{a=1}^{k-1}(w-1)\mathbb{I}_i\left(\bigcup_{j=a}^{k-1}\hat{s}_j \cup \hat{c}_a\right)$$
$$+ \sum_{i=1}^{M}\sum_{a=2}^{k-1}\mathbb{I}_i\left(\bigcup_{j=a}^{k-1}\hat{s}_j\right) + M. \quad (52)$$

All the admissible policies cannot schedule any sensors when $k \leq T$ because they are in sleep mode during that time. Therefore, the value of (52) is same for all the admissible policies when $k \leq T+1$. Below we prove the theorem by considering two complementary cases, i.e., $M < T+1$ and $M \geq T+1$.

1) For the case $M < T+1$, when $k \geq T+M+1$, we can know the change rule of the value of AoI-penalty for all the sensors during one transmission cycle is $\{1, 2, \ldots, T+1\}$, which is the optimal change rule for all sensors. From Corollary 2, we can get

$$\Delta^* = \Delta^{\mathrm{MW}} = \frac{T}{2} + 1. \quad (53)$$

2) For the another case $M \geq T+1$, according to the proof of Theorem 3, there always exists one sensor that is scheduled at the beginning of each slot when $T+1 \leq k$, and the order is $(1, 2, \ldots, M, 1, 2, \ldots)$. Therefore, the MW policy maximizes the number of deliveries and for every $\pi \in \Pi$ and we have

$$\left|\bigcup_{j=1}^{k}\hat{s}_j^{\mathrm{MW}}\right| \geq \left|\bigcup_{j=1}^{k}\hat{s}_j^{\pi}\right| \quad (54)$$

where $|s|$ means the number of elements of set s and MW represents the MW policy. In addition, we can get

$$\left|\bigcup_{j=a}^{k}\hat{s}_j^{\mathrm{MW}}\right| \geq \left|\bigcup_{j=a}^{k}\hat{s}_j^{\pi}\right|, \quad a \in \{1, 2, \ldots, k\}. \quad (55)$$

Also, the MW policy maximizes the number of sensors that are in sleep mode. If the intersection of $\bigcup_{j=a}^{k}\hat{s}_j^{\mathrm{MW}}$ and $\hat{c}_a$ is a empty set, the MW policy maximizes the number of elements of the union of them. If the intersection of $\bigcup_{j=a}^{k}\hat{s}_j^{\mathrm{MW}}$ and $\hat{c}_a$ is not an empty set, the number of elements is M, which is the maximum admissible value. So we can know

$$\left|\bigcup_{j=a}^{k}\hat{s}_j^{\mathrm{MW}} \cup \hat{c}_a^{\mathrm{MW}}\right| \geq \left|\bigcup_{j=a}^{k}\hat{s}_j^{\pi} \cup \hat{c}_a^{\pi}\right|, \quad a \in \{1, 2, \ldots, k\}. \quad (56)$$

Equation (54) shows the MW policy minimizes the first term in RHS of (52). The last term in RHS of (52) is constant. The second term in RHS of (52) can be written as

$$\sum_{i=1}^{M}\sum_{a=1}^{k-1}(w-1)\mathbb{I}_i\left(\bigcup_{j=a}^{k-1}\hat{s}_j \cup \hat{c}_a\right)$$
$$= \sum_{a=1}^{k-1}(w-1)\left[M - \left|\bigcup_{j=a}^{k-1}\hat{s}_j \cup \hat{c}_a\right|\right] \quad (57)$$

and the third term in RHS of (52) can be written as

$$\sum_{i=1}^{M}\sum_{a=2}^{k-1}\mathbb{I}_i\left(\bigcup_{j=a}^{k-1}\hat{s}_j\right) = \sum_{a=2}^{k-1}\left[M - \left|\bigcup_{j=a}^{k-1}\hat{s}_j\right|\right]. \quad (58)$$

According to (55)–(58), the MW policy minimizes (52) in every slot k in this case. From Corollary 2, we can get $\Delta^* = \Delta^{\mathrm{MW}} = f(M)$ which completes the proof.

References

[1] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The Industrial Internet of Things (IIoT): An analysis framework," *Comput. Ind.*, vol. 101, pp. 1–12, Oct. 2018.

[2] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[4] L. Hu, Z. Chen, Y. Dong, Y. Jia, L. Liang, and M. Wang, "Status update in IoT networks: Age of information violation probability and optimal update rate," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11329–11344, Jul. 2021.

[5] J. Li, Y. Zhou, and H. Chen, "Age of information for multicast transmission with fixed and random deadlines in IoT systems," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8178–8191, Sep. 2020.

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

[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, Nov. 2017.

[8] H. Tang, P. Ciblat, J. Wang, M. Wigger, and R. Yates, "Age of information aware cache updating with file-and age-dependent update durations," in *Proc. 18th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOPT)*, 2020, pp. 1–6.

[9] M. Bastopcu and S. Ulukus, "Age of information for updates with distortion: Constant and age-dependent distortion constraints," 2019. [Online]. Available: arXiv:1912.13493.

[10] R. Han, J. Wang, L. Bai, J. Liu, and J. Choi, "Age of information and performance analysis for UAV-aided IoT systems," *IEEE Internet Things J.*, early access, Jan. 13, 2021, doi: 10.1109/JIOT.2021.3051361.

[11] Y. Nakayama and K. Maruta, "Age-of-information-based host selection for mobile user provided networks," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 672–683, Jan. 2021.

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

[13] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2637–2650, Dec. 2018.

[14] C. Chau, F. Qin, S. Sayed, M. H. Wahab, and Y. Yang, "Harnessing battery recovery effect in wireless sensor networks: Experiments and analysis," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1222–1232, Sep. 2010.

[15] J.-B. Seo, "On minimizing energy consumption of duty-cycled wireless sensors," *IEEE Commun. Lett.*, vol. 19, no. 10, pp. 1698–1701, Oct. 2015.

[16] D.-T. Le, T. Le-Duc, T. Kong, V. V. Zalyubovskiy, and H. Choo, "Exploring energy-latency trade-off for broadcast in duty-cycled wireless sensor networks," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2016, pp. 98–101.

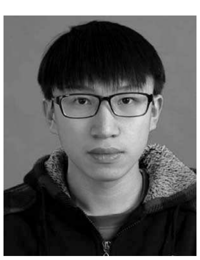
[17] S. K. Kaul and R. D. Yates, "Age of information: Updates with priority," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 2644–2648.

[18] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "On the age of information with packet deadlines," *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6419–6428, Sep. 2018.

[19] C. Li, S. Li, Y. Chen, Y. T. Hou, and W. Lou, "Minimizing age of information under general models for IoT data collection," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2256–2270, Oct.–Dec. 2020.

[20] B. Sombabu and S. Moharir, "Age-of-information based scheduling for multi-channel systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4439–4448, Jul. 2020.

[21] P. R. Jhunjhunwala, B. Sombabu, and S. Moharir, "Optimal AoI-aware scheduling and cycles in graphs," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1593–1603, Mar. 2020.
[22] C. Li, S. Li, Y. Chen, Y. T. Hou, and W. Lou, "AoI scheduling with maximum thresholds," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2020, pp. 436–445.
[23] I. Kadota and E. Modiano, "Minimizing the age of information in wireless networks with stochastic arrivals," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 1173–1185, Mar. 2021.
[24] B. Yin, S. Zhang, and Y. Cheng, "Application-oriented scheduling for optimizing the age of correlated information: A deep-reinforcement-learning-based approach," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8748–8759, Sep. 2020.
[25] Y. P. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *IEEE Trans. Mobile Comput.*, vol. 19, no. 12, pp. 2903–2915, Dec. 2020.
[26] H. H. Yang, A. Arafa, T. Q. S. Quek, and H. V. Poor, "Optimizing information freshness in wireless networks: A stochastic geometry approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 6, pp. 2269–2280, Jun. 2021.
[27] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor, "Age-minimal transmission for energy harvesting sensors with finite batteries: Online policies," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 534–556, Jan. 2020.
[28] Y. Dong, P. Fan, and K. B. Letaief, "Energy harvesting powered sensing in IoT: Timeliness versus distortion," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10897–10911, Nov. 2020.
[29] H. Tang, J. Wang, L. Song, and J. Song, "Minimizing age of information with power constraints: Multi-user opportunistic scheduling in multi-state time-varying channels," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 854–868, May 2020.
[30] A. M. Bedewy, Y. Sun, R. Singh, and N. B. Shroff, "Low-power status updates via sleep–wake scheduling," *IEEE/ACM Trans. Netw.*, early access, May 31, 2021, doi: 10.1109/TNET.2021.3081102.
[31] D. Sinha and R. Roy, "Scheduling status update for optimizing age of information in the context of industrial cyber-physical system," *IEEE Access*, vol. 7, pp. 95677–95695, 2019.
[32] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Age and value of information: Non-linear age case," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 326–330.
[33] Y. Sun and B. Cyr, "Sampling for data freshness optimization: Non-linear age functions," *J. Commun. Netw.*, vol. 21, no. 3, pp. 204–219, 2019.
[34] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," in *Synthesis Lectures on Communication Networks*, vol. 3. San Rafael, CA, USA: Morgan Claypool, 2010, pp. 1–211.
[35] S. M. Ross *et al.*, *Stochastic Processes*, vol. 2. New York, NY, USA: Wiley, 1996.
[36] K. F. Ramadan, M. Dessouky, M. Abd-Elnaby, and F. E. Abd El-Samie, "Energy-efficient dual-layer MAC protocol with adaptive layer duration for WSNs," in *Proc. 11th Int. Conf. Comput. Eng. Syst. (ICCES)*, 2016, pp. 47–52.



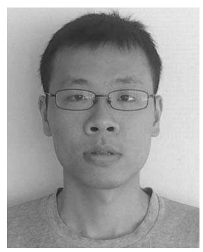
Jia Wang (Graduate Student Member, IEEE) received the B.E. degree in control science and engineering from Harbin Engineering University, Harbin, China, in 2019. He is currently pursuing the master's degree with the School of Automation, Southeast University, Nanjing, China.

His research interests include transmission scheduling of wireless networks.

Xianghui Cao (Senior Member, IEEE) received the B.S. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively.

From 2012 to 2015, he was a Senior Research Associate with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA. He is currently a Professor with the School of Automation, Southeast University, Nanjing, China. His current research interests include cyber–physical systems, wireless network performance analysis, wireless networked control, and network security.

Prof. Cao was a recipient of the Best Paper Runner-Up Award from ACM MobiHoc in 2014 and the First Prize of Natural Science Award of Ministry of Education of China in 2017. He also serves as an Associate Editor for *ACTA Automatica Sinica* and IEEE/CAA JOURNAL OF AUTOMATICA SINICA.



Bo Yin (Student Member, IEEE) received the B.E. degree in electronic information engineering and the M.E. degree in electronic science and technology from Beihang University, Beijing, China, in 2010 and 2013, respectively, and the Ph.D. degree in computer engineering from Illinois Institute of Technology, Chicago, IL, USA, in 2020.

His research interests include network resource allocation, age of information, and ML-based network optimization.

Yu Cheng (Senior Member, IEEE) received the B.E. and M.E. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2003.

He is currently a Full Professor with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA. His research interests include wireless network performance analysis, information freshness, machine learning, and network security.

Prof. Cheng received the Best Paper Award at QShine 2007 and IEEE ICC 2011, the Runner-Up Best Paper Award at ACM MobiHoc 2014, the National Science Foundation CAREER Award in 2011, and the IIT Sigma Xi Research Award in the junior faculty division in 2013. He has served as the Symposium Co-Chair for IEEE ICC and IEEE GLOBECOM and the Technical Program Committee Co-Chair for IEEE/CIC ICCC 2015, ICNC 2015, and WASA 2011. He was the Founding Vice Chair of the IEEE ComSoc Technical Subcommittee on Green Communications and Computing. He was an IEEE ComSoc Distinguished Lecturer from 2016 to 2017. He is an Associate Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE INTERNET OF THINGS JOURNAL, and IEEE WIRELESS COMMUNICATIONS.