Optimal Sleep Scheduling for Energy-Efficient AoI Optimization in Industrial Internet of Things

Xianghui Cao, *Senior Member, IEEE*, Jia Wang, *Student Member, IEEE*, Yu Cheng, *Senior Member, IEEE*, and Jiong Jin, *Member, IEEE*

***Abstract*—Keeping sensor data fresh is desired for Industrial Internet of Things (IIoT), especially, in real-time monitoring applications. However, this may require sensors always in active mode and, thus, incur low energy efficiency. In this article, we consider that a wireless sensor monitors a dynamical system and reports real-time measurements to a processing center through an unreliable wireless channel. We study the problem of optimizing the sensor data freshness in terms of Age of Information (AoI) while saving energy by scheduling the sensor to sleep when needed. The problem is formulated as a Markov decision process that takes both AoI and energy consumption into account, to which we theoretically prove that the optimal scheduling policy forms a cyclic sleep–wake pattern. The optimal sleep period is also analyzed. Simulation results demonstrate that the proposed scheduling policy outperforms other existing policies.**

***Index Terms*—Age of Information (AoI), energy, Markov decision process (MDP), optimization, sleep scheduling.**

I. Introduction

In INDUSTRIAL Internet of Things (IIoT) applications such as real-time monitoring and control, the freshness of sensor data is of great importance [1], [2]. For example, in remote monitoring applications, it has been shown that the fresher data the remote estimator can receive, the smaller the state estimation error the estimator could achieve [3]. Another example is the federated learning applications in IIoT, where the freshness of training data becomes particularly important when data cannot remain structurally similar across time [4].

Although delay is one of the most commonly used metrics of packet-wise transmission performance in networks, it does not accurately reflect the data freshness [5], [6]. Recently, the notion of Age of Information (AoI) has been proposed, which tracks the time elapsed since the generation of the latest received data from the perspective of the receiver, as a new metric of data freshness [3], [4], [5], [6], [7], [8], [9]. In the literature, a number of studies have been devoted to minimizing AoI for fresh data gathering over wireless networks [7], [8], e.g., the Max-Weight policy [7], Whittle's Index policy [7], and SQRT-Weight policy [8]. However, many existing studies assume that the sensors are always in active mode and ready for data transmission once scheduled [7], [8].

Intuitively, the wireless sensor may try to seize every opportunity to transmit data in order to minimize the AoI. However, such an always-on working mode may cause significant energy waste for the sensor during its idle time, and, hence, may be even unaffordable for a resource-constrained sensor [10]. In IIoT, energy efficiency is an important issue, and improving the energy efficiency of wireless devices and prolonging their lifetime becomes increasingly important as wireless technologies are expected to gain more penetration in future IIoT [11], [12].

In this article, we consider a class of real-time monitoring applications of IIoT with the aim at enhancing the sensor data freshness as characterized by AoI while reducing the energy consumption of the sensor. Motivated by the working mode of duty-cycling sensors [13], we allow the sensor to sleep in order to save energy. Many wireless technologies have specific mechanisms similar to the sleeping mode. For example, traditional IEEE 802.11 specifies the power-saving mode, and the recently released IEEE 802.11ax standard further introduces the target wake time agreement to save energy [14]. Other wireless technologies, such as WirelessHART and ZigBee, also employ low-power modes to save energy.

Our basic idea is to save the sensor's energy by switching it off when needed without sacrificing much AoI. By doing so, we are able to achieve a balance between information freshness about the dynamic process at the remote processing center and the sensor's energy consumption. Specifically, the sensor saves energy when sleeping without sending any data and improves AoI only when it is in active mode. Taking both switching energy and the energy for being active of the sensor into account, we formulate an optimization problem of the sensor's sleep scheduling with the objective being a combination of a generic AoI function and the total energy consumption. Based on a Markov decision process (MDP), we theoretically prove that the optimal solution yields a cyclic working pattern, i.e., in each cycle, the sensor first sleeps for a fixed period and then wakes up and keeps active until successfully sending a data

Manuscript received 29 September 2022; revised 15 December 2022; accepted 1 January 2023. Date of publication 6 January 2023; date of current version 23 May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 92067111; in part by the Defense Industrial Technology Development Program under Grant JCKY2020206B068; in part by the Natural Science Foundation of Jiangsu Province under Grant BK20202006; and in part by the Zhongying Young Scholars Project. The work of Yu Cheng was supported by NSF under Grant CNS-1816908 and Grant CNS-2008092. *(Corresponding author: Xianghui Cao.)*

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

Yu Cheng is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: cheng@iit.edu).

Jiong Jin is with the School of Science, Computing and Engineering Technologies, Swinburne University of Technology, Melbourne, VIC 3122, Australia (e-mail: jiongjin@swin.edu.au).

Digital Object Identifier 10.1109/JIOT.2023.3234582

packet to the processing center. We compare our scheduling policy with other existing policies, and the results demonstrate that our policy outperforms the other in terms of AoI–energy tradeoff. The main contributions are summarized as follows.

1) We formulate an optimal sensor sleep scheduling problem that trades off between AoI and sensor's energy consumption. In the literature, although there are some studies on AoI minimization in IIoT recently [6], [7], very few of them shed light on the energy-efficiency issue of AoI optimization by using sensor sleep mode.
2) We theoretically prove that the optimal sleep scheduling policy leads to a cycling pattern.
3) We propose an algorithm to find the optimal sleep period. Moreover, for cases that the AoI function is linear, we derive an explicit expression of the optimal sleep period.

The remainder of this article is organized as follows. Section II summarizes some related work. Section III introduces the system model and formulates the optimization problem. Section IV derives the optimal sleep scheduling policy and proposes ways to find the optimal sleep period. Section V presents the simulation results, while Section VI concludes this article.

Notations: $\oplus$ denotes the XOR operator. $\mathbb{E}(\cdot)$ denotes the mathematical expectation. Let $\mathrm{tr}(\cdot)$ represent the trace of a matrix. $\Pr(\cdot)$ denotes the probability of an event. Let $\mathbb{N}$ be the set of nonnegative integers. Denote by $\mathbb{1}_{\text{event}}$ the indicator function, which equals to 1 if event is true and 0 if otherwise.

II. Related Work

Recently, AoI has attracted more and more attention in applications where data freshness matters. For example, the problems of when to sample the data and in what order to process the data in order to optimize AoI are studied in [15] and [16]. In time-slotted systems, transmission scheduling for minimizing AoI is studied in [7] and [17].

Due to the limited energy of wireless sensors in many IIoT systems, it is of great importance to save energy when minimizing AoI. For example, under the constraint that the average energy consumption of the sensor cannot exceed a given value, a policy that can reduce AoI is proposed to choose proper sensors to send their updates [18]. In [19], the problem of data sending scheduling in order to minimize AoI and energy is studied. Another thread of research toward energy-efficient AoI minimization is to consider rechargeable sensors [20], [21]. In [20], the source node needs the energy to sample and send data, while the destination node, with power supply, can transfer wireless energy to charge the source node. Then, a joint sampling, charging, and updating policy is proposed to minimize AoI. As for the case where energy arrives randomly, in order to reduce AoI, Zhou et al. [21] proposed optimal offline policies and efficient online policies to schedule the transmitter whether to send the update when it arrives.

In the literature, a few works have been devoted to energy-efficient AoI minimization with sensors that are allowed to sleep to save energy [22], [23]. In [22], the sensors will sleep if they find the channel is busy. And with the constraint of energy consumption, the optimal sleep parameters are derived to minimize peak AoI. In [23], a new AoI-penalty function is proposed to characterize the data eagerness for sensors that wake up after sleeping for a certain time. Then, a Max-Weight-based sensor scheduling policy is proposed to minimize the sensors' AoI-penalty. The above studies leverage sensors' sleep mode to save energy, which, however, lacks a theoretical explanation whether and to what extent the sensor can benefit in terms of AoI by employing sleep mode.

TABLE I
Definitions of Key Notations

Notation	Definition
k	The discrete-time step
$s(k)$	The working mode (active or sleep) of the sensor during step k
$u(k)$	The scheduling decision (to keep or to switch mode) at the beginning of step k
p	The successful transmission probability
$\gamma(k)$	Indicator of whether the remote center receives the sensor data in k
$t_g(k)$	The generation time of the freshest data the remote center received by the end of step $k-1$
$\Delta(k)$	Age of information (AoI) at the beginning of step k
$f(\cdot)$	The AoI function
E_a, E_s	The one-step energy consumption of the sensor for being in active and sleep, respectively
$E_{\text{on}}, E_{\text{off}}$	The energy consumption for mode switching
$C(k)$	The total one-step energy consumption of the sensor
$J(k), \bar{J}^*$	The total cost function and its optimal value
π, π^*	The sleep scheduling policy and its optimal solution
$R(\cdot,\cdot,\cdot), V_t(\cdot,\cdot)$	The reward and value functions, respectively
$\Delta_{\text{active}}, \Delta_{\text{sleep}}$	The thresholds in the optimal scheduling policy π^*
T, T^*	The sleep period and its optimal value

III. System Model

The main notations used throughout this article are summarized in Table I.

We consider a real-time monitoring system in which a wireless sensor measures the dynamical state of a physical process and sends the measurement data through an unreliable wireless channel to a remote data processing center (DPC) [24], [25]. For example, in [24], a sensor sends real-time measurements of a 2-DOF (degree of freedom) serial flexible joint robot to a controller for monitoring and control purposes, where the dynamics of the robot is modeled as follows:

$$x(k) = Ax(k-1) + B\theta(k-1) + w(k-1) \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ is the system state, $\theta \in \mathbb{R}^{n_c}$ is the control input, and A and B are coefficient matrices with proper dimensions. w is the system noise which is Gaussian with zero mean and covariance matrix Σ. $k \in \{1, 2, \ldots, K\}$ is the index of the discrete-time steps. Note that (1) is only an example of the physical process. This article focuses on a generic form of AoI, which does not rely on any specific forms of the dynamic process model. For other processes under monitoring, as long as the corresponding AoI function evolves as in (5) below, the results obtained in this article remain valid.

When the wireless sensor is in active mode, it measures the state of the physical process at the beginning of the current step and sends the measurement to the DPC. Whereas,

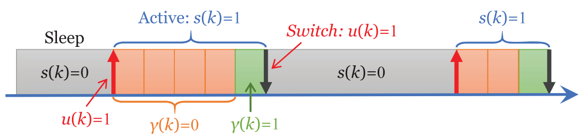


Fig. 1. Schematic of the binary variables.

it keeps inactive to save energy in sleep mode. At the beginning of every step, the sensor chooses whether to switch the current mode or not. Let $s(k) \in \{0, 1\}$ denote the working mode of the sensor during step k (after switching) with $s(k)$ equals to 1 indicating active mode and 0 otherwise. Denote by $u(k) \in \{0, 1\}$ the switching decision at the beginning of step k[1] such that $u(k) = 0$ means keeping the mode while $u(k) = 1$ means otherwise. Therefore, the working mode of the sensor during step k is determined by

$$s(k) = s(k-1) \oplus u(k). \tag{2}$$

The relationship between $s(k)$ and $u(k)$ is shown in Fig. 1.

We assume that each sensor packet transmission can be completed within one step and that the DPC replies the sensor an acknowledgment upon successfully receiving the sensor packet. The transmissions from the sensor are over an unreliable wireless channel with the probability of a successful data transmission as $p \in (0, 1]$. Since the acknowledgment from the energy-rich DPC is short in length and can be sent with high power, we assume its transmission is reliable [7]. Let $\gamma(k) = 1$ indicate that the DPC successfully receives the sensor's measurement in k and $\gamma(k) = 0$ otherwise. We have

$$\mathbb{E}\big[\gamma(k)\big] = p\mathbb{E}[s(k)] = p\mathbb{E}[s(k-1) \oplus u(k)]. \tag{3}$$

A. Optimization Problem

According to [5], the AoI at the DPC at the beginning of step k is defined as the time difference between the generation time of the freshest measurement packet that the DPC receives from the sensor. Let $t_g(k)$ denote the generation time of the freshest measurement the DPC receives from the sensor by the end of step $k-1$. Denote by $\Delta(k) = k - t_g(k)$ the AoI at the beginning of step k (the definition of AoI in [7]). If the sensor is in active mode and transmits the measurement successfully during step k (i.e., $\gamma(k) = 1$), at the beginning of the next step, $t_g(k+1) = k$; otherwise, $t_g(k+1) = t_g(k)$. Hence, we have

$$\Delta(k+1) = \begin{cases} 1, & \text{if } \gamma(k) = 1 \\ \Delta(k) + 1, & \text{otherwise.} \end{cases} \tag{4}$$

We define an AoI function $f(\Delta)$ that evolves as follows [15]:

$$f(\Delta(k+1)) = \begin{cases} f(1), & \text{if } \gamma(k) = 1 \\ f(\Delta(k) + 1), & \text{otherwise} \end{cases} \tag{5}$$

where $f(\cdot)$ is assumed nondecreasing. The AoI function is a metric of how AoI impacts the system performance. Taking the real-time monitoring system in (1), for example, the AoI function can be defined as [3]

$$f(\Delta) = \sum_{i=0}^{\Delta-1} \operatorname{tr}\Big(\big(A^T\big)^i A^i \Sigma\Big) \tag{6}$$

where Σ is the covariance of the system noise w. Equation (6) represents the mean-squared error of the state estimation performance in terms of AoI at the controller. In communication systems, typical AoI functions are defined as $f(\Delta) = \Delta$ (e.g., [5], [6], and [7]) and $f(\Delta) = e^{\Delta}$ (e.g., [15]). Notice that the above AoI functions satisfy the assumption that they are nondecreasing.

B. Problem Formulation

Denote the energy consumption of the sensor as $E_a > 0$ and $E_s > 0$ ($E_a > E_s$) at every step when it is in active and sleep modes, respectively. Besides, in order to save energy, the sensor should avoid switching its working mode too frequently. As a consequence, we denote the energy consumption of the sensor for waking up (from sleep to active) and turning off (from active to sleep) as $E_{\text{on}} > 0$ and $E_{\text{off}} > 0$, respectively. Then, the total energy consumption during step k is

$$\begin{aligned} C(k) &= E_a[s(k-1) \oplus u(k)] + E_s[1 - s(k-1) \oplus u(k)] \\ &\quad + E_{\text{on}} u(k)[1 - s(k-1)] + E_{\text{off}} u(k) s(k-1). \end{aligned} \tag{7}$$

For example, if $s(k-1) = 0$ and $u(k) = 1$, the sensor wakes up from the sleep mode and keeps in active during k, and, hence, $C(k) = E_a + E_{\text{on}}$.

Then, the total cost function during step k is set as a weighted combination of both AoI function and energy consumption as follows:

$$J(k) = (1-\lambda) f\big(\Delta(k)\big) + \lambda C(k) \tag{8}$$

where $\lambda \in [0, 1]$ represents the weight of energy consumption in the optimization objective. On the one hand, minimizing AoI would require the sensor to keep active trying to send its data as quickly as possible, which results in high energy consumption. On the other hand, saving energy by letting the sensor sleep may miss some data sending opportunities and, hence, sacrifices the AoI. The larger the value of λ, the more the sensor prefers to sleep to save energy; otherwise, it prefers to reduce the value of AoI if its energy is rich. For the above problem to be meaningful, hereafter we assume that $0 \le \lambda < 1$. Since the AoI function can be application specific, its magnitude may be different from the energy consumption. Therefore, in order to analyze them together, both the AoI function and the energy consumption should be normalized.

Let $\pi = [u(1), \ldots, u(K)]$ be a sleep scheduling policy that determines whether the sensor sleep or awake and Π be the set of all admissible policies. Without loss of generality, we set $t_g(1) = 0$, $\Delta(1) = 1$, and $s(1) = 0$. Then, our optimization problem can be formulated as follows:

$$\min_{\pi \in \Pi} \bar{J} = \lim_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[J(k)] \tag{9a}$$

$$\text{s.t.} \quad u(k) \in \{0, 1\} \; \forall k. \tag{9b}$$

Let $\bar{J}^*$ be the optimal solution.

[1]In practice, if in sleep mode, the sensor may not be able to make switching decisions. A viable way can be that the sensor decides a sleep period and sets a wake-up timer accordingly before it sleeps. In this article, for the ease of problem formulation, we assume that the sensor can decide whether to wake up or not when sleeping. However, our proposed optimal policy does not need this assumption.

IV. Optimal Scheduling Policy

In this section, we first reformulate Problem (9) as an MDP, based on which we find a structural property of the optimal policy. Then, we derive the optimal scheduling policy and analytically characterize its performance.

A. MDP Formulation of Problem (9)

At the beginning of step k, define the state and action of the sensor as $(\Delta(k), s(k-1))$ and $u(k)$, respectively, with the action space $\{0, 1\}$. If the sensor was in sleep mode, i.e., $s(k-1) = 0$, based on the dynamics of $\Delta(k)$ as given in (4), one can see that $\Delta(k+1) = \Delta(k) + 1$ if $u(k) = 0$. If the sensor chooses to switch its working mode, i.e., $u(k) = 1$ when $s(k-1) = 0$, the change of AoI relies on the channel state during step k in terms of that $\Delta(k+1)$ will drop to 1 if the transmission is successful (with probability p); otherwise, $\Delta(k+1) = \Delta(k) + 1$. Therefore, the one-step probability transfer function from current state $(\Delta, 0)$ to a new state (Δ', s') at the beginning of the next step under action u can be summarized as follows:

$$\Pr\left(\Delta', s' | \Delta, 0\right) = \begin{cases} 1, & \text{if } u = s' = 0 \text{ and } \Delta' = \Delta + 1 \\ p, & \text{if } u = s' = 1 \text{ and } \Delta' = 1 \\ 1-p, & \text{if } u = s' = 1 \text{ and } \Delta' = \Delta + 1 \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Similarly, we can derive the probability transfer function when the sensor was previously in active mode as

$$\Pr\left(\Delta', s' | \Delta, 1\right) = \begin{cases} 1, & \text{if } u = 1, s' = 0 \text{ and } \Delta' = \Delta + 1 \\ p, & \text{if } u = 0, s' = 1 \text{ and } \Delta' = 1 \\ 1-p, & \text{if } u = 0, s' = 1 \text{ and } \Delta' = \Delta + 1 \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

In view of (9), define the reward as

$$\begin{aligned} R(\Delta, s, u) = &(1-\lambda)f(\Delta) + \lambda E_a(s \oplus u) \\ &+ \lambda E_s[1 - (s \oplus u)] + \lambda E_{\text{on}} u(1-s) \\ &+ \lambda E_{\text{off}} us \end{aligned} \tag{12}$$

where we have used (7) and (8). Then, the value function $V_t(\Delta, s)$ can be given as follows:

$$V_{t+1}(\Delta, s) = \min_{\pi \in \Pi}\left\{R(\Delta, s, u) + \beta \mathbb{E}\left[V_t\left(\Delta', s'\right)\right]\right\} \tag{13}$$

where $\beta \in (0, 1)$ is the discount factor. The above value iteration can start at any initial value function V_0, and for convenience, we set $V_0(\Delta, s) = (1-\lambda)f(\Delta)$. Then, $V_t(\Delta, s)$ converges to the optimal value function $V^*(\Delta, s) = \lim_{t\to\infty} V_t(\Delta, s)$ for any $\Delta \geq 0$ and $s \in \{0, 1\}$.

When the sensor was previously in sleep mode, submitting (10) and (11) into (13), we have

$$\begin{aligned} V_{t+1}(\Delta, 0) &= \min_{u\in\{0,1\}}\left\{(1-\lambda)f(\Delta) + \lambda u E_a + \lambda(1-u)E_s + \lambda u E_{\text{on}} \right.\\ &\quad \left. + \beta \mathbb{E}\left[V_t\left(\Delta', s'\right)\right]\right\} \\ &= \min_{u\in\{0,1\}}\{(1-\lambda)f(\Delta) + \lambda u E_a + \lambda(1-u)E_s + \lambda u E_{\text{on}} \\ &\quad + \beta(1-u)V_t(\Delta+1, 0) + \beta \text{up} V_t(1, 1) \\ &\quad + \beta u(1-p)V_t(\Delta+1, 1)\} \\ &= \min_{u\in\{0,1\}}\left\{(1-u)L_{1,t}(\Delta) + uL_{2,t}(\Delta)\right\} \\ &= \min\left\{L_{1,t}(\Delta), L_{2,t}(\Delta)\right\} \qquad (14) \\ &= L_{1,t}(\Delta) + \min_{u\in\{0,1\}}\left\{u\left[L_{2,t}(\Delta) - L_{1,t}(\Delta)\right]\right\} \\ &= L_{1,t}(\Delta) + \min_{u\in\{0,1\}}\left\{uL_{\text{sleep},t}(\Delta)\right\} \qquad (15) \end{aligned}$$

where we have used (10) in deriving the second equality. In the above

$$L_{1,t}(\Delta) \triangleq (1-\lambda)f(\Delta) + \lambda E_s + \beta V_t(\Delta+1, 0) \tag{16}$$

$$\begin{aligned} L_{2,t}(\Delta) \triangleq &(1-\lambda)f(\Delta) + \lambda E_{\text{on}} + \lambda E_a \\ &+ \beta p V_t(1, 1) + \beta(1-p)V_t(\Delta+1, 1) \end{aligned} \tag{17}$$

$$\begin{aligned} L_{\text{sleep},t}(\Delta) &\triangleq L_{2,t}(\Delta) - L_{1,t}(\Delta) \\ &= -\beta\left[V_t(\Delta+1, 0) - (1-p)V_t(\Delta+1, 1)\right] \\ &\quad + \lambda(E_{\text{on}} + E_a - E_s) + \beta p V_t(1, 1). \end{aligned} \tag{18}$$

Similarly, when the sensor was previously active

$$\begin{aligned} &V_{t+1}(\Delta, 1) \\ &= \min_{u\in\{0,1\}}\left\{(1-\lambda)f(\Delta) + \lambda(1-u)E_a + \lambda u E_s + \lambda u E_{\text{off}} \right.\\ &\quad \left. + \beta \mathbb{E}\left[V_t\left(\Delta', s'\right)\right]\right\} \\ &= \min_{u\in\{0,1\}}\left\{uL_{3,t}(\Delta) + (1-u)L_{4,t}(\Delta)\right\} \\ &= \min\left\{L_{3,t}(\Delta), L_{4,t}(\Delta)\right\} \qquad (19) \\ &= L_{4,t}(\Delta) + \min_{u\in\{0,1\}}\left\{uL_{\text{active},t}(\Delta)\right\} \qquad (20) \end{aligned}$$

where we have used (11) in deriving the second equality and

$$L_{3,t}(\Delta) \triangleq L_{1,t}(\Delta) + \lambda E_{\text{off}} \tag{21}$$

$$L_{4,t}(\Delta) \triangleq L_{2,t}(\Delta) - \lambda E_{\text{on}} \tag{22}$$

$$L_{\text{active},t}(\Delta) \triangleq L_{3,t}(\Delta) - L_{4,t}(\Delta). \tag{23}$$

In the sequel, we shall drop the subscript t in $L_{i,t}$, $L_{\text{sleep},t}$, and $L_{\text{active},t}$ to indicate their converged values as $t \to \infty$.

B. Optimal Policy

Based on the above, we can derive the following properties of the value function.

Lemma 1: $\forall \Delta \geq 0$ and $\forall n \geq 0$, the following inequalities hold:

$$1)\ V^*(\Delta+n, s) \geq V^*(\Delta, s) \tag{24}$$

$$\begin{aligned} 2)\ &V^*(\Delta+n, 0) - V^*(\Delta, 0) \\ &\geq V^*(\Delta+n, 1) - V^*(\Delta, 1) \end{aligned} \tag{25}$$

$$\begin{aligned} 3)\ &V^*(\Delta+n, 0) - (1-p)V^*(\Delta+n, 1) \\ &\geq V^*(\Delta, 0) - (1-p)V^*(\Delta, 1). \end{aligned} \tag{26}$$

Proof: The proof is provided in the Appendix. ■

Next, we derive a threshold structure of the optimal policy.

Lemma 2: $\forall \Delta \geq 0$, there exist $\Delta_{\text{active}} \leq \Delta_{\text{sleep}} \leq \infty$ such that the optimal scheduling policy π^* has a threshold structure in terms of that

$$u^*(\Delta, s) = \begin{cases} \mathbb{1}_{\Delta \geq \Delta_{\text{sleep}}}, & \text{if s} = 0 \\ \mathbb{1}_{\Delta \leq \Delta_{\text{active}}}, & \text{if s} = 1 \end{cases} \tag{27}$$

where $u^*(\Delta, s)$ represents the optimal scheduling decision when the sensor is in state (Δ, s).

Proof: According to (15) and (20), the optimal scheduling decision depends on the sign of $L_{\text{sleep}}(\Delta)$ and $L_{\text{active}}(\Delta)$. When $L_{\text{sleep}}(\Delta) > 0$ or $L_{\text{active}}(\Delta) > 0$, (15) and (20) suggest that the corresponding optimal scheduling decision is $u^* = 0$. Similarly, if $L_{\text{sleep}}(\Delta) < 0$ or $L_{\text{active}}(\Delta) < 0$, the corresponding optimal scheduling decision is $u^* = 1$. When $L_{\text{sleep}}(\Delta) = 0$ or $L_{\text{active}}(\Delta) = 0$, the corresponding optimal scheduling decision u^* is derived as below by considering whether $L_{\text{sleep}}(\Delta) = 0$ or $L_{\text{active}}(\Delta) = 0$ have nonnegative solutions.

First, suppose that $L_{\text{sleep}}(\Delta) = 0$ (or $L_{\text{active}}(\Delta) = 0$) has nonnegative solutions. Let Δ_{sleep} be the maximum nonnegative solution of the following:

$$\begin{aligned} L_{\text{sleep}}(\Delta) = {} & \lambda E_{\text{on}} - \lambda E_s + \lambda E_a + \beta p V^*(1,1) \\ & - \beta\big[V^*(\Delta+1,0) - (1-p)V^*(\Delta+1,1)\big] \\ = {} & 0. \end{aligned} \tag{28}$$

Consider that the sensor was previously in sleep mode, i.e., $s = 0$. According to (26), $L_{\text{sleep}}(\Delta)$ is nonincreasing in Δ. Therefore, combining (28) and $L_{\text{sleep}}(\Delta_{\text{sleep}}) = 0$, we have the following inequalities hold for any $n > 0$:

$$L_{\text{sleep}}\big(\Delta_{\text{sleep}} + n\big) \le L_{\text{sleep}}\big(\Delta_{\text{sleep}}\big) = 0 \tag{29}$$

$$L_{\text{sleep}}\big(\Delta_{\text{sleep}} - n\big) \ge L_{\text{sleep}}\big(\Delta_{\text{sleep}}\big) = 0. \tag{30}$$

From (29), one can see that the optimal scheduling decision is $u^* = 1$ when the current AoI is greater than Δ_{sleep}. Similarly, from (30), one can see that the optimal scheduling decision is $u^* = 0$ when the current AoI is less than Δ_{sleep}. Notice that if the nonnegative solutions of (28) are not unique, the solutions must be a continuous interval, say $[\underline{\Delta}_{\text{sleep}}, \Delta_{\text{sleep}}]$, which is because $L_{\text{sleep}}(\Delta)$ is nonincreasing in Δ. In this case, without affecting its optimality, u^* can be set as 1 if the current AoI is greater than Δ_{sleep} and 0 if otherwise. Above all, we obtain the first line of (27).

When the sensor was previously in active mode, i.e., $s = 1$, in a similar argument, we obtain the second line of (27), where Δ_{active} is the minimum nonnegative solution of the following:

$$\begin{aligned} L_{\text{active}}(\Delta) = {} & \beta V^*(\Delta+1,0) - \beta(1-p)V^*(\Delta+1,1) \\ & + \lambda E_{\text{off}} + \lambda E_s - \lambda E_a - \beta p V^*(1,1) \\ = {} & 0. \end{aligned} \tag{31}$$

Besides, according to (18) and (23)

$$L_{\text{active}}(\Delta) + L_{\text{sleep}}(\Delta) = \lambda E_{\text{on}} + \lambda E_{\text{off}} \ge 0. \tag{32}$$

At the point Δ_{active} where $L_{\text{active}}(\Delta_{\text{active}}) = 0$, $L_{\text{sleep}}(\Delta_{\text{active}})$ should be nonnegative according to (32), i.e.,

$$L_{\text{sleep}}(\Delta_{\text{active}}) > 0 = L_{\text{sleep}}\big(\Delta_{\text{sleep}}\big). \tag{33}$$

Therefore, $\Delta_{\text{sleep}} \ge \Delta_{\text{active}}$ due to that $L_{\text{sleep}}(\Delta)$ is nonincreasing in Δ.

Next, we discuss the cases when $L_{\text{sleep}}(\Delta) = 0$ or $L_{\text{active}}(\Delta) = 0$ has no nonnegative solution. For ease of exposition, in what follows, when we say one of $L_{\text{sleep}}(\Delta) = 0$ and $L_{\text{active}}(\Delta) = 0$ has no nonnegative solution while the other has nonnegative solution, we mean that the other has a unique solution. For the cases when it has multiple solutions, our analysis below remain valid for the reason similar to the above.

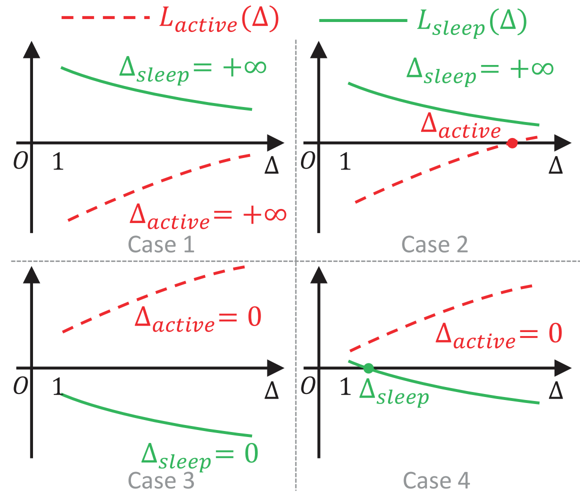


Fig. 2. Four cases where $L_{\text{sleep}}(\Delta) = 0$ or $L_{\text{active}}(\Delta) = 0$.

As shown in Fig. 2, there are four cases for that $L_{\text{sleep}}(\Delta) = 0$ or $L_{\text{active}}(\Delta) = 0$ has no nonnegative solution.

1) *Case 1:* If $L_{\text{sleep}}(\Delta) > 0$ and $L_{\text{active}}(\Delta) < 0 \ \ \forall \Delta \ge 0$. Since $\forall \Delta \ge 0$, $L_{\text{sleep}}(\Delta) > 0$, the optimal scheduling decision is $u^* = 0$ when the sensor was in sleep mode according to (15). Similarly, since $\forall \Delta \ge 0$, $L_{\text{active}}(\Delta) < 0$, the optimal scheduling decision is $u^* = 1$ when the sensor was in active mode. Therefore, (27) holds if letting $\Delta_{\text{sleep}} = \Delta_{\text{active}} = \infty$.
2) *Case 2:* If $\forall \Delta \ge 0$, $L_{\text{sleep}}(\Delta) > 0$ while $L_{\text{active}}(\Delta) = 0$ has a nonnegative solution Δ_{active}. As aforementioned, the optimal scheduling decision is $u^* = 0$ when the sensor was previously in sleep mode, which validates the first line of (27) by letting $\Delta_{\text{sleep}} = \infty$. Meanwhile, the second line of (27) holds for the same reason above (31).
3) *Case 3:* If $L_{\text{sleep}}(\Delta) < 0$ and $L_{\text{active}}(\Delta) > 0 \ \forall \Delta \ge 0$. Similar to case 1, (27) holds if letting $\Delta_{\text{sleep}} = \Delta_{\text{active}} = 0$.
4) *Case 4:* If $\forall \Delta \ge 0$, $L_{\text{active}}(\Delta) > 0$ while $L_{\text{sleep}}(\Delta) = 0$ has a nonnegative solution Δ_{sleep}. Similar to case 2, (27) holds if letting $\Delta_{\text{active}} = 0$.

In sum, the optimal scheduling decision of u^* is given in (27) for some $\Delta_{\text{active}} \le \Delta_{\text{sleep}} \le \infty$. ■

The threshold structure in Lemma 2 can be interpreted as follows. When the sensor was previously in sleep mode with low energy consumption, the increasing AoI becomes dominating the cost function $J(k)$. When the value of AoI becomes excessively large, the sensor switches to the active mode and spends some energy to transmit data in order to make AoI drop. The switching happens at the point Δ, which is the smallest integer greater than Δ_{sleep} with $L_{\text{sleep}}(\Delta_{\text{sleep}}) = 0$. This can be viewed as that the expected change of the value function V due to mode switching, i.e., $L_2(\Delta) - L_1(\Delta)$ as in (18), is beneficial to reducing the total cost J. The scheduling decision threshold Δ_{active} can be interpreted

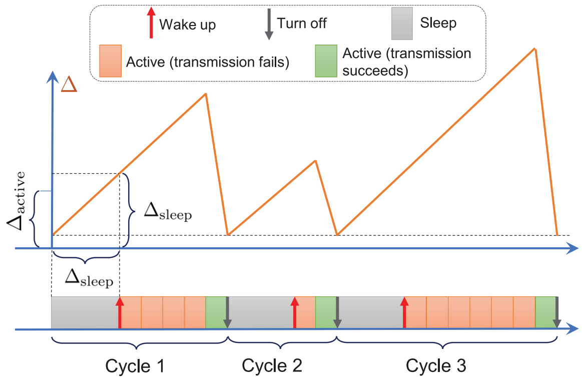


Fig. 3. Illustration of the sleep–wake working pattern.

similarly. In extreme cases, e.g., $\lambda = 1$ (i.e., only energy consumption matters in J) or $p = 0$ (i.e., the wireless channel between the sensor and the remote DPC is blocked), the sensor may keep in sleep mode, which can be viewed as $\Delta_{\text{sleep}} = \infty$.

Based on Lemma 2, we can further obtain the working pattern of the sensor.

Theorem 1: Under the optimal sleep scheduling policy, the sensor works in a cycling pattern as follows: in each cycle, it first sleeps for a fixed period $T \in \mathbb{N}$, and then wakes up and keeps active until successfully sending a data packet. After that, a new cycle begins and it switches to sleep mode again.

Proof: As has been assumed, initially $\Delta = 1$ and $s = 0$. According to Lemma 2, $\Delta \geq 1$ and $\Delta_{\text{sleep}} \geq \Delta_{\text{active}}$. Since Δ_{sleep} and Δ_{active} can be any of $\{0, 1, \ldots, \infty\}$, we prove the theorem by dividing the problem into the following cases.

1) *If $\Delta_{active} \leq \Delta_{sleep} \leq 1$:* The sensor will keep active all the time, which is equivalent to that the sleep period T is equal to 0.
2) *If $\Delta_{sleep} = \infty$:* The sensor will sleep all the time, and the sleep period T is equal to ∞.
3) *If $\Delta_{active} \leq 1$ and $1 < \Delta_{sleep} < \infty$:* The sensor will stay in sleep mode until the AoI gets greater than Δ_{sleep}. Then, the sensor will switch to active mode and start to transmit the measurement. After that, the sensor will keep active all the time and the sleep period T is equal to 0.
4) *Otherwise:* The sensor will stay in sleep mode until the AoI is greater than Δ_{sleep}. Then, the sensor will switch to active mode and start to transmit the measurement. The AoI will drop to 1 when the transmission is successful. After that, the sensor will turn to sleep since $\Delta_{\text{active}} \geq 1$.

Above all, the sensor will repeat the above working process to form a cycle, where $T = \Delta_{\text{sleep}}$. ■

An example of the sleep–wake working pattern is depicted in Fig. 3.

C. Performance Analysis

We may wonder about the performance comparison of AoI and energy consumption between the sensor with sleep scheduling policy and the nonsleeping sensor in [7]. Since the sensor will be nonsleeping when the sleep period $T = 0$, we only discuss the case where $T \geq 1$. Let φ_{AoI} and φ_{energy} denote the ratio of time-average AoI function and time-average energy consumption between the sensor with sleep scheduling policy and nonsleeping policy, respectively.

Lemma 3: The ratio of time-average AoI and time-average energy consumption between the sensor under sleep scheduling policy ($T \geq 1$) and the nonsleeping sensor are, respectively

$$\varphi_{\text{AoI}} = \frac{\sum_{\Delta=1}^{T} f(\Delta) + \sum_{\Delta=1}^{\infty}(1-p)^{\Delta-1} f(T+\Delta)}{(pT+1)\sum_{\Delta=1}^{\infty}(1-p)^{\Delta-1} f(\Delta)} \tag{34}$$

$$\varphi_{\text{energy}} = \frac{\tilde{E}}{(pT+1)E_a} + \frac{E_s}{E_a} \tag{35}$$

where $\tilde{E} = E_a - E_s + p\text{Sign}(T)(E_{\text{on}} + E_{\text{off}})$ and $\text{Sign}(T)$ equals to 1 if $T > 0$ and 0 otherwise.

Proof: First, let us consider the sensor under the proposed sleep–wake policy. In each cycle, the AoI Δ evolves as follows: it starts at 1 and grows to T during the sleeping period. After that, Δ grows to $T + \ell$, where ℓ is the number of transmission trials for the sensor to successfully transmit a packet. Thus, the deliveries of the sensor data form a renewal process [26], and the number ℓ follows a geometric distribution with $\Pr(\ell = j) = p(1-p)^{j-1}$ and $\mathbb{E}[\ell] = 1/p$. Therefore, the average energy consumption of the sleep–wake sensor is

$$\frac{E_a\mathbb{E}[\ell] + TE_s + E_{\text{on}} + E_{\text{off}}}{T + \mathbb{E}[\ell]} = \frac{\tilde{E}}{pT+1} + E_s \tag{36}$$

and the averaged AoI function is

$$\begin{aligned}
\frac{\mathbb{E}\left[\sum_{\Delta=1}^{T+\ell} f(\Delta)\right]}{T+\mathbb{E}[\ell]} &= \frac{p\mathbb{E}\left[\sum_{\Delta=1}^{T+\ell} f(\Delta)\right]}{pT+1} \\
&= \frac{p}{pT+1}\sum_{\ell=1}^{\infty}\sum_{\Delta=1}^{T+\ell} p(1-p)^{\ell-1} f(\Delta) \\
&= \frac{p^2}{pT+1}\left[\sum_{\Delta=1}^{T}\sum_{\ell=1}^{\infty}(1-p)^{\ell-1} f(\Delta)\right. \\
&\qquad \left. + \sum_{\Delta=1}^{\infty}\sum_{\ell=\Delta}^{\infty}(1-p)^{\ell-1} f(T+\Delta)\right] \\
&= \frac{p}{pT+1}\left[\sum_{\Delta=1}^{T} f(\Delta) + \sum_{\Delta=1}^{\infty}(1-p)^{\Delta-1} f(T+\Delta)\right].
\end{aligned} \tag{37}$$

Then, for the nonsleeping sensor, the average energy consumption is E_a and the averaged AoI function is

$$p\mathbb{E}\left[\sum_{\Delta=1}^{\ell} f(\Delta)\right] = p\sum_{\Delta=1}^{\infty}(1-p)^{\Delta-1} f(\Delta). \tag{38}$$

Thus, (34) and (35) are proved. ■

Remark 1: It is difficult to analyze (34) for a generic AoI function. If we consider the special AoI function $f(\Delta) = \Delta$, (34) can be simplified as $\varphi_{\text{AoI}} = (1/2)(p + pT + 1 + ([1-p]/[pT+1])) \geq 1$. On the other hand, since usually E_s, E_{on}, and E_{off} are much smaller than E_a, (35) reduces to $\varphi_{\text{energy}} \approx (1/[pT+1]) \leq 1$. Therefore, by applying the sleep–wake policy, the sensor is able to balance between AoI and energy consumption. Moreover, by using an optimal sleep period T, the sleep–wake sensor is able to achieve better performance in terms of $\bar{J}$ than the nonsleeping sensor.

Further, we may be interested in the optimal value of the sleep period T^* in the sleep–wake working pattern. Below we first show that the optimal T^* is finite in normal cases and then we derive an expression of T^* for a commonly used AoI function [6], [7]: $f(\Delta) = \Delta$.

Theorem 2: If $\lambda \neq 1$ and $p \neq 0$, and the AoI function increases at a speed faster than $\alpha > 0$, i.e., $f(\Delta+1) - f(1) \geq \alpha\Delta$, then the optimal sleep period is upper bounded as

$$T^* \leq \bar{T}^* \triangleq \frac{\lambda}{\alpha p(1-\lambda)}\tilde{E} - 1. \tag{39}$$

Additionally, if $\tilde{E} < 2\alpha p(1/\lambda - 1)$, $T^* = 0$.

Proof: By (36) and (37), the objective of Problem (9) is

$$\bar{J}(T) = \frac{(1-\lambda)p}{pT+1}\left[\sum_{\Delta=1}^{T} f(\Delta) + \sum_{\Delta=1}^{\infty}(1-p)^{\Delta-1} f(T+\Delta)\right] + \lambda\left(\frac{\tilde{E}}{pT+1} + E_s\right). \tag{40}$$

Then, let us examine the monotonicity of $\bar{J}(T)$. We have

$$\begin{aligned}
&\bar{J}(T+1) - \bar{J}(T) \\
&\propto (1-\lambda)p\left[(pT+1)\sum_{\Delta=1}^{T+1} f(\Delta) - (pT+p+1)\sum_{\Delta=1}^{T} f(\Delta)\right. \\
&\quad + (pT+1)\sum_{\Delta=1}^{\infty}(1-p)^{\Delta-1} f(T+1+\Delta) \\
&\quad \left. - (pT+p+1)\sum_{\Delta=1}^{\infty}(1-p)^{\Delta-1} f(T+\Delta)\right] - \lambda p\tilde{E}
\end{aligned} \tag{41}$$

where $\propto$ means "proportional to." After some rearrangements, the right-hand side of the above becomes

$$\begin{aligned}
&(1-\lambda)p^2\left[p(T+1)\sum_{\Delta=1}^{\infty}(1-p)^{\Delta-1} f(T+1+\Delta)\right. \\
&\quad \left. - \sum_{\Delta=1}^{T+1} f(\Delta) - \frac{\lambda}{(1-\lambda)p}\tilde{E}\right] \\
&\geq (1-\lambda)p^2\left[(T+1)f(T+2) - \sum_{\Delta=1}^{T+1} f(\Delta) - \frac{\lambda}{(1-\lambda)p}\tilde{E}\right] \\
&\geq (1-\lambda)p^2\left[f(T+2) - f(1) - \frac{\lambda}{(1-\lambda)p}\tilde{E}\right] \\
&\geq (1-\lambda)p^2\left[\alpha(T+1) - \frac{\lambda}{(1-\lambda)p}\tilde{E}\right].
\end{aligned} \tag{42}$$

Therefore, when $T \geq \bar{T}^*$, the objective $\bar{J}(T)$ becomes monotonically increasing, which means the optimal T^* exists and is below $\bar{T}^*$. ■

From the above, one can see that when $\lambda = 1$ or $p = 0$, the optimal sleep period is ∞, which means the sensor better sleep all the time to save energy. For a special case of the AoI function, the optimal sleep period can be obtained as follows.

Corollary 1: In case if the AoI function is defined as $f(\Delta) = \Delta$, under the sleep–wake working pattern, the optimal cost is

$$\bar{J}^*(T) = \frac{1-\lambda}{2}T + \frac{(1-p)(1-\lambda) + 2\lambda p\tilde{E}}{2p(1+pT)} + \frac{(1-\lambda)(1+p)}{2p} + \lambda E_s \tag{43}$$

and the optimal sleep period T^* that minimizes $\bar{J}^*(T)$ is

$$T^* = \underset{T\in\{0,\lceil\tilde{T}\rceil,\lfloor\tilde{T}\rfloor\}}{\arg\min}\ \bar{J}^*(T) \tag{44}$$

where

$$\tilde{T} = \max\left\{1, \frac{1}{p}\left(\sqrt{\frac{2p\lambda}{1-\lambda}\tilde{E} + 1 - p} - 1\right)\right\}. \tag{45}$$

Proof: First, (43) can be directly obtained by submitting $f(\Delta)$ into (40). When $T \geq 1$, by letting the derivative of $J(T)$ with respect to T [i.e., $J'(T)$] be zero, we obtain

$$\hat{T} = \frac{1}{p}\left(\sqrt{\frac{2p\lambda}{1-\lambda}\tilde{E} + 1 - p} - 1\right)$$

which is not necessarily an integer. If $\hat{T} \leq 1$, then $J'(1) \geq 0$ and $J(T)$ increases monotonically when $T \geq 1$. In this case, the optimal sleep period T^* is 1. If $\hat{T} > 1$, then $J'(T) \leq 0$ when $1 \leq T \leq \hat{T}$ and $J'(T) > 0$ when $T > \hat{T}$. Therefore, $J(T)$ decreases monotonically when $1 \leq T \leq \hat{T}$ and increases monotonically when $T > \hat{T}$. In this case, the optimal sleep period T^* should be the closest integer around $\hat{T}$. ■

Remark 2: If $f(\Delta) = \Delta$, one can see that, when E_a, E_{on}, and E_{off} increase, the optimal sleep period T^* increases so that the sensor can sleep longer to save energy. However, when E_{on} or E_{off} become very large, the sensor will spend a lot of mode-switching energy. In such a situation, the sensor may be unwilling to spend extra energy to switch its working mode and, hence, will keep active all the time. This corresponds to the case of $T^* = 0$ in Corollary 1.

Remark 3: Above we have characterized the optimal sleep period T^* in a special case. For a generic AoI function $f(\Delta)$, it is difficult to find a generic expression of T^*. Therefore, we propose Algorithm 1 to search the optimal sleep period T^*. The computation complexity of this algorithm is $O(\bar{T}^*\zeta)$ where ζ is the time complexity of calculating the objective function in (40).

Algorithm 1: Searching the Optimal Sleep Period T^*

Input: E_a, E_s, E_{on}, E_{off}, p, λ, $f(\Delta)$, $\bar{T}^*$;
$T^* \leftarrow 0$, $T \leftarrow 0$, $J_{\min} \leftarrow \infty$;
for $T = 1$ **to** $\bar{T}^*$ **do**
    Calculate J by (40) ;
    if $J < J_{\min}$ **then**
        $T^* \leftarrow T$, $J_{\min} \leftarrow J$;
    end
end

V. Simulation Results

In our simulations, we consider remotely estimating the state of a dynamical system (1) with parameters

$$A = \begin{bmatrix} 1 & 0.2 \\ -0.1 & 1.2 \end{bmatrix},\quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\quad \Sigma = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}. \tag{46}$$

The AoI function is chosen as (6) which represents the mean-squared error of the remote estimation performance [3]. According to the parameters of TR1001 in [27], the default settings of the following parameters are $E_a = 3.6 \times 10^{-4} J$, $E_s = 0.0015 \times 10^{-4} J$, $E_{\text{on}} = 0.252 \times 10^{-4} J$, $E_{\text{off}} = 0.0283 \times 10^{-4} J$, $p = 0.5$, $\lambda = 0.5$, and $K = 2 \times 10^4$.

A. Performance of the Proposed Sleep Scheduling Policy

First, we evaluate the performance of the proposed sleep scheduling policy by simulations and compare it with four other existing policies—Greedy policy [6], optimal stationary randomized (OSR) policy [7], duty-cycle (DC) policy [13], and dynamic programming (DP)-based policy [28]. The Greedy policy makes decisions according to the cost it brings. Specifically, during step k, the Greedy policy considers the energy cost during step k and the AoI during step $k+1$. Therefore, the sensor changes its working mode from sleep to active when

$$\begin{aligned}\lambda(E_{\text{on}} + E_a) + (1-\lambda)\big[pf(1) + (1-p)f(\Delta(k)\ +1)\big] \\ \le (1-\lambda)f(\Delta(k)+1) + \lambda E_s\end{aligned} \tag{47}$$

and changes its working mode from active to sleep when

$$\begin{aligned}\lambda(E_{\text{off}} + E_s) + (1-\lambda)f(\Delta(k)+1) \\ \le \lambda E_a + (1-\lambda)\big[pf(1) + (1-p)f(\Delta(k)\ +1)\big].\end{aligned} \tag{48}$$

The OSR policy switches the sensor's working mode with a fixed probability $\beta \in [0, 1]$ at every step. The deliveries of the measurement form a renewal process. Then, we can establish an optimization problem to minimize the average cost in (9) and get the optimal probability β^*, which forms the OSR policy. The DC policy makes the sensor switch the working mode at a fixed time interval [29], in which we set the sleep and active periods of equal length which is calculated by (44) for a fair comparison. We apply the DP method as in [28] to optimally solve the optimization problem (9) and the solution forms the DP policy. Then, we conduct Monte Carlo simulations to evaluate the proposed sleep scheduling policy and the above four existing policies in terms of the average cost in (9). The results reported in the following figures are averages of 3×10^4 independent simulation runs.

Fig. 4 shows the results with different $E_{\text{on}} \in \{0.152, 0.252, 0.352, \ldots, 1.052\}$. Fig. 5 shows the results with different $E_a \in \{3.5, 3.6, 3.7, \ldots, 4.4\}$. Fig. 6 shows the results with different $p \in \{0.3, 0.4, 0.5 \ldots, 1\}$. From Figs. 4–6, we can find that the proposed optimal sleep scheduling policy has the best average cost performance. And the performance of the optimal sleep scheduling policy is almost coincident with that of the DC policy. This also reflects the correctness of Theorem 1 and Corollary 1. The DC requires backtracking every time a decision is made, which leads to a very high time complexity and may rise the so-called "dimension disaster" problem [28].

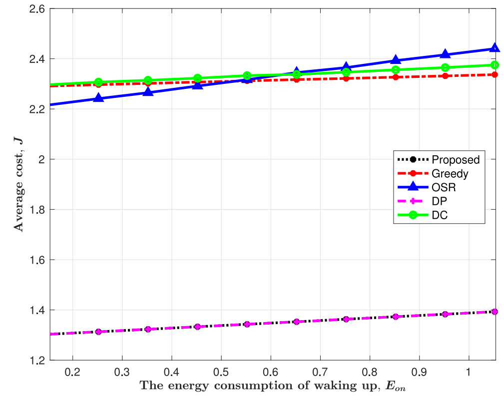


Fig. 4. Performance comparisons under different E_{on}.

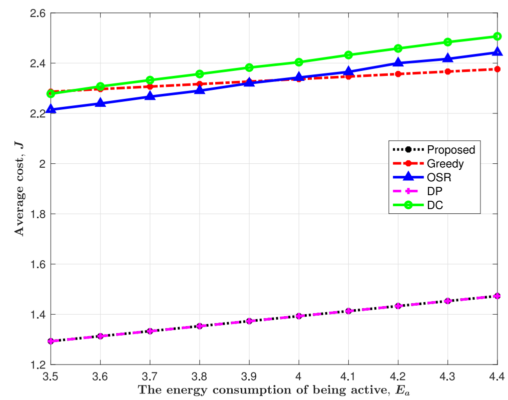


Fig. 5. Performance comparisons under different E_a.

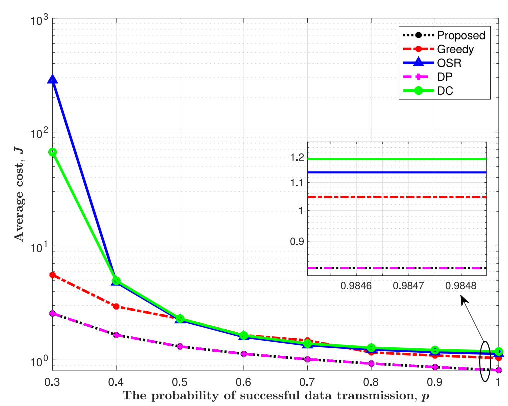


Fig. 6. Performance comparisons under different p.

Moreover, in practice, the sensor if in sleep mode may not be able to make scheduling decisions. The DC policy may be not applicable in this situation. Therefore, a viable method can be that the sensor decides a sleep period and sets a wake-up timer before it sleeps. The proposed optimal sleep scheduling policy meets the above condition and it only needs to calculate the optimal sleep period T^* in advance. The sensor only needs to work according to the preset pattern when running, and no extra calculation is needed when making decisions. We evaluate the performance of our policy when the function is the most commonly used one [15]: $f(\Delta) = \Delta$. As shown in Fig. 7, the optimal sleep scheduling policy also has the best

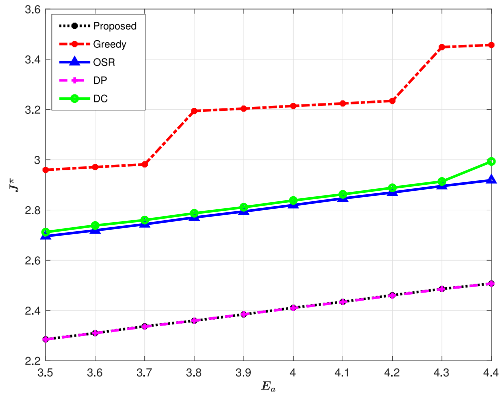


Fig. 7. Performance comparisons under different E_a with $f(\Delta) = \Delta$.

performance. Notice that the curve of the Greedy policy has some inflection points. The reason is as follows. Based on (47) and (48), it is easy to deduce that, under the Greedy policy and with $f(\Delta) = \Delta$, the sensor also works in a cyclic pattern similar to that in Theorem 1, except that the sleeping period is $\lceil(\lambda/[p(1-\lambda)])(E_a + E_{\text{on}} - E_s)\rceil$. That is, the inflection points occur mainly due to the nonsmoothness of this sleeping period. In contrast, in our sleep scheduling policy, the optimal sleeping period is used which is derived from solving the original optimization problem.

B. Performance Comparison With Event-Based Policy

We compare the sleep scheduling policy and a typical Event-Based policy as proposed in [30] for reducing measurement transmissions in remote state estimation systems. In the Event-Based policy [30], during every step, the sensor generates an independent random variable $\phi(k)$ which is uniformly distributed over the interval $[0, 1]$. When the sensor is in sleep mode, it will switch to active mode when $\phi(k) > \exp(-(1/2)\Delta(k)^2 Y)$, where Y is a positive parameter. When the sensor is in active mode, it will switch to sleep mode when $\phi(k) < \exp(-(1/2)\Delta(k)^2 Y)$. Fig. 8 shows the performance of the optimal sleep scheduling policy and the Event-Based policy. With the increase of Y, the Event-Based policy will make the sensor stay in active mode for a longer time, resulting in the decrease of AoI function and the increase of energy consumption. Fig. 8(c) shows the average cost comparisons of the optimal sleep scheduling policy and the Event-Based policy. We can see that the optimal sleep scheduling policy outperforms the other one. In addition, since the event-based policy requires the sensor to make decisions at every step, the sensor might have to wake up at the beginning of every step to decide whether to remain in active or sleep again. This may incur extra energy cost. In contrast, in our proposed policy, the sensor only needs to maintain a wakeup clock once sleeps and wake up when the clock counts down to 0.

C. Optimal Sleep Period T^*

Furthermore, we analyze the optimal sleep period T^* by simulations under the proposed sleep scheduling policy. Fig. 9 demonstrates the value of the optimal sleep period T^* under

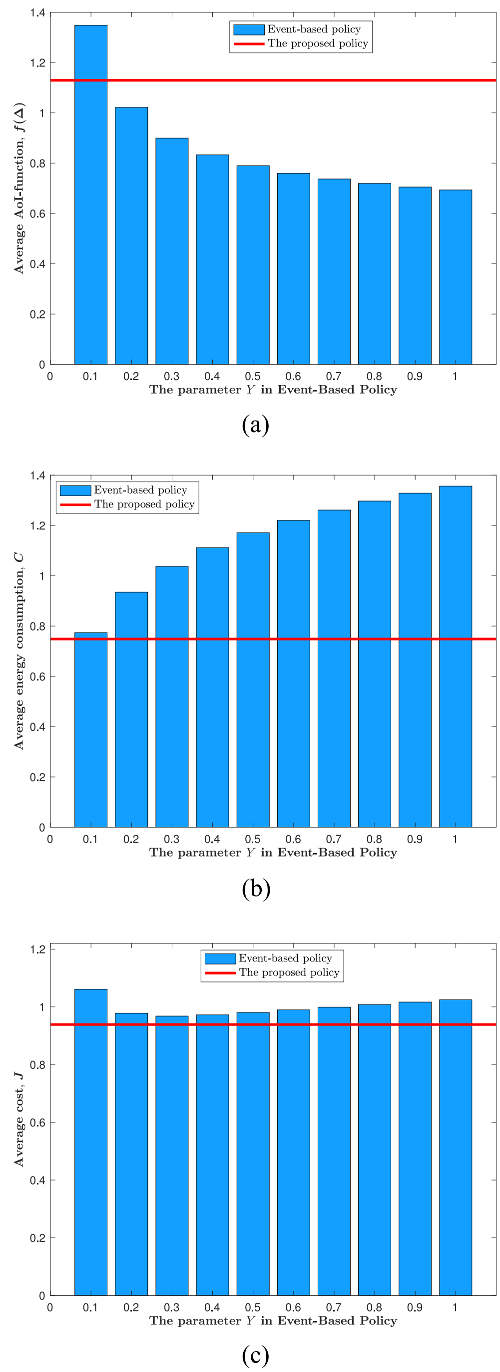


Fig. 8. Performance comparison with event-based policy [30]. (a) Average AoI function. (b) Average energy consumption. (c) Average energy consumption.

different E_{on} and E_a. For a fix E_a, the optimal sleep period increases when the value of E_{on} increases. However, when E_{on} exceeds a certain value, the optimal sleep period drops to 0. This is because the sensor becomes unwilling to spend more energy to wake up as E_{on} is high, and, hence, it keeps active all the time and $T^* = 0$. On the other hand, for a fixed E_{on}, the figure shows that the larger the E_a is, the longer the optimal sleep period will be. This is reasonable because when the energy consumption for staying in active mode grows larger, the sensor prefers to sleep for a longer time to save

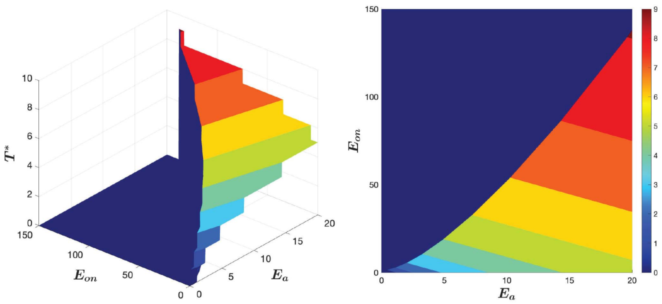


Fig. 9. Optimal value of sleep period T^* under different E_{on} and E_a.

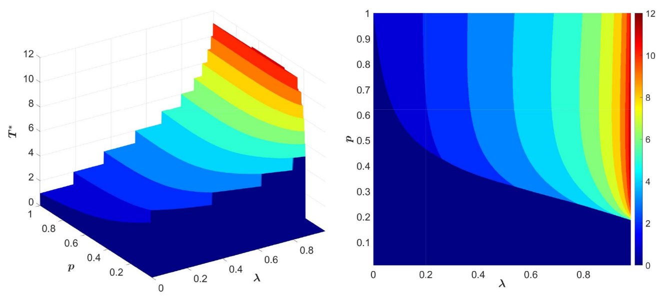


Fig. 10. Optimal value of sleep period T^* under different p and λ.

energy. Fig. 10 demonstrates the value of the optimal sleep period T^* under different p and λ. We can observe that the larger the λ is, the longer the optimal sleep period becomes. This is because the larger the weight of energy consumption is, the more time the sensor will spend in sleeping to save energy. In addition, with a smaller successful data transmission rate p, the sensor is expected to spend more time in active in order to successfully deliver a packet, leading to a higher energy consumption. Thus, it needs to sleep less to reduce AoI by using a shorter sleep period, as shown in Fig. 10.

D. Comparison With Nonsleeping Sensor

Then, we compare the performance of AoI and energy consumption between the sensor with the optimal sleep scheduling policy and the nonsleeping sensor [7]. Fig. 11 demonstrates the comparison results with $\lambda \in \{0.1, 0.2, \ldots, 0.9\}$. When $\lambda = 0.1$, the optimal sleep period is 0 for the sleep–wake sensor, so the AoI and energy cost of the sleep–wake sensor are the same as the nonsleeping sensor. When λ is larger, the energy cost is more important and the optimal sleep period is larger. Although the optimal sleep–wake sensor sacrifices the freshness of the data, it saves a lot of energy. Therefore, introducing sleep mode for the sensor and adopting the optimal sleep scheduling policy in this article are good attempts to balance AoI and energy consumption.

VI. Conclusion

Although the freshness of the data is quite important, we have to consider the energy consumption of the sensor in IIoT. In order to provide fresh information while reducing the energy consumption of the sensor, we introduce two working modes for the sensor: 1) sleep mode and 2) active mode. The sensor can switch its mode at the beginning of every step. Then, we establish an AoI and energy consumption optimization problem. The problem is formulated into an MDP and we reveal the optimal scheduling policy follows a kind of threshold structure. Furthermore, we find the optimal scheduling policy forms a cyclic sleep–wake working pattern. Then, we theoretically obtain the ratio of average AoI function and average energy consumption between the sleep–wake sensor and the nonsleeping sensor. Later, we theoretically derive the optimal sleep period when the AoI function is linear and we propose an algorithm to find the optimal sleeping period for the common AoI function. In the simulation results, we can find that the proposed optimal sleep scheduling policy has the best performance. Moreover, the simulation demonstrates the introduction of sleep mode and the optimal sleep scheduling policy saves a lot of energy, which is a good way to balance the freshness of data and energy consumption.

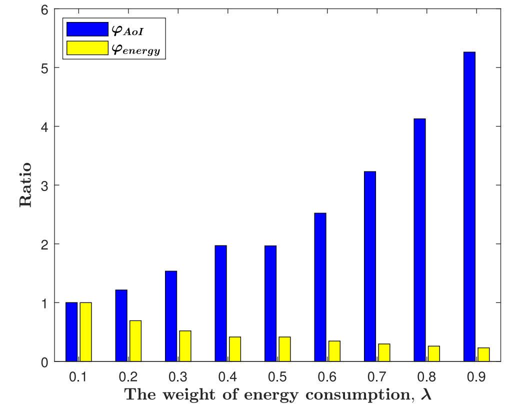


Fig. 11. Performance comparison with the nonsleeping sensor scenario.

Appendix
Proof of Lemma 1

Proof: First, we prove (24) by induction. Initially when $t = 0$, we choose $V_0(\Delta, s) = (1 - \lambda)f(\Delta)$. Then, we can get $V_0(\Delta + n, s) = (1 - \lambda)f(\Delta + n)$. Since $f(\Delta)$ is nondecreasing, it holds that $V_0(\Delta + n, s) \geq V_0(\Delta, s)$ $\forall\Delta \geq 0$ and $\forall n \geq 0$. Next, assuming that $V_t(\Delta + n, s) \geq V_t(\Delta, s)$ $\forall\Delta \geq 0$ and $\forall n \geq 0$, we need to prove that this inequality also holds for $t + 1$. Because $V_t(\Delta + n, 0) \geq V_t(\Delta, 0)$, $L_{1,t}(\Delta + n) \geq L_{1,t}(\Delta)$, and $L_{2,t}(\Delta + n) \geq L_{2,t}(\Delta)$, i.e., all the two terms in the minimum operation of (14) are nondecreasing, we have $V_{t+1}(\Delta + n, 0) \geq V_{t+1}(\Delta, 0)$ $\forall\Delta \geq 0$ and $\forall n \geq 0$. Similarly, we can get $V_{t+1}(\Delta + n, 1) \geq V_{t+1}(\Delta, 1)$ $\forall\Delta \geq 0$ and $\forall n \geq 0$. As a consequence, $V_{t+1}(\Delta + n, s) \geq V_{t+1}(\Delta, s)$ $\forall\Delta \geq 0$ and $\forall n \geq 0$. By induction, $\forall t \in \mathbb{N}$

$$V_t(\Delta + n, s) \geq V_t(\Delta, s)\ \forall\Delta \geq 0\ \forall n \geq 0. \tag{49}$$

Thus, when $t \to \infty$, we have $V^*(\Delta + n, s) \geq V^*(\Delta, s)$ $\forall\Delta \geq 0$ and $\forall n \geq 0$.

Second, we prove (25) by induction. Similar to the above, we choose $V_0(\Delta, s) = (1 - \lambda)f(\Delta)$. Then, we can get $V_0(\Delta + n, 0) - V_0(\Delta, 0) = V_0(\Delta + n, 1) - V_0(\Delta, 1)$ $\forall\Delta \geq 0$ and $\forall n \geq 0$. Next, we assume that the above inequality holds for iteration t and we need to prove the inequality for $t+1$, i.e.,

$$V_{t+1}(\Delta+n,0)-V_{t+1}(\Delta,0) \geq V_{t+1}(\Delta+n,1)-V_{t+1}(\Delta,1). \tag{50}$$

Based on (21) and (22), it is obvious that $L_{1,t}(\Delta) \leq L_{3,t}(\Delta)$ and $L_{2,t}(\Delta) \geq L_{4,t}(\Delta)$. Moreover, according to the proof of (24), L_1, L_2, L_3, and L_4 are nondecreasing in Δ. From iteration t on, we have the following inequality for any $n \geq 0$ and $\Delta \geq 0$:

$$\begin{aligned} L_{3,t}(\Delta+n)-L_{3,t}(\Delta) &= L_{1,t}(\Delta+n)-L_{1,t}(\Delta) \\ &= (1-\lambda)(f(\Delta+n)-f(\Delta)) \\ &\quad + \beta V_t(\Delta+n+1,0) - \beta V_t(\Delta+1,0) \\ &\geq (1-\lambda)(f(\Delta+n)-f(\Delta)) \\ &\quad + \beta V_t(\Delta+n+1,1) - \beta V_t(\Delta+1,1) \\ &\geq (1-\lambda)(f(\Delta+n)-f(\Delta)) \\ &\quad + \beta(1-p)V_t(\Delta+n+1,1) \\ &\quad - \beta(1-p)V_t(\Delta+1,1) \\ &= L_{2,t}(\Delta+n)-L_{2,t}(\Delta) \\ &= L_{4,t}(\Delta+n)-L_{4,t}(\Delta) \end{aligned} \tag{51}$$

where we have used (49) in deriving the first inequality. To prove (50), we divide the problem into the following 20 cases according to the property of L_1, L_2, L_3, and L_4. The following discussions are valid for any $n \geq 0$ and any $\Delta \geq 0$.

1) *Case 1:* If $L_{1,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{1,t}(\Delta+n) \leq L_{3,t}(\Delta+n) \leq L_{4,t}(\Delta+n) \leq L_{2,t}(\Delta+n)$, based on (14) and (19), we have

$$\begin{aligned} V_{t+1}(\Delta+n,0)-V_{t+1}(\Delta,0) &= L_{1,t}(\Delta+n)-L_{1,t}(\Delta) \\ V_{t+1}(\Delta+n,1)-V_{t+1}(\Delta,1) &= L_{3,t}(\Delta+n)-L_{3,t}(\Delta). \end{aligned}$$

Then, $V_{t+1}(\Delta+n,0) - V_{t+1}(\Delta,0) = V_{t+1}(\Delta+n,1) - V_{t+1}(\Delta,1)$ by (51). Therefore, (50) holds in this case.

2) *Case 2:* If $L_{1,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{1,t}(\Delta+n) \leq L_{4,t}(\Delta+n) \leq L_{3,t}(\Delta+n) \leq L_{2,t}(\Delta+n)$, we can get

$$\begin{aligned} V_{t+1}(\Delta+n,0)-V_{t+1}(\Delta,0) &= L_{1,t}(\Delta+n)-L_{1,t}(\Delta) \\ &= L_{3,t}(\Delta+n)-L_{3,t}(\Delta) \\ V_{t+1}(\Delta+n,1)-V_{t+1}(\Delta,1) &= L_{4,t}(\Delta+n)-L_{3,t}(\Delta) \\ &\leq L_{3,t}(\Delta+n)-L_{3,t}(\Delta). \end{aligned}$$

where we have used $L_{4,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$. Hence, (50) holds in this case.

3) *Case 3:* If $L_{1,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{1,t}(\Delta+n) \leq L_{4,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 2, (50) holds.

4) *Case 4:* If $L_{1,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{3,t}(\Delta+n) \leq L_{2,t}(\Delta+n)$, similar to case 2, (50) holds.

5) *Case 5:* If $L_{1,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 2, (50) holds.

6) *Case 6:* If $L_{1,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, we can get

$$\begin{aligned} V_{t+1}(\Delta+n,0)-V_{t+1}(\Delta,0) &= L_{2,t}(\Delta+n)-L_{1,t}(\Delta) \\ &= L_{4,t}(\Delta+n)-L_{3,t}(\Delta)+\lambda E_{\text{on}}+\lambda E_{\text{off}} \\ &\geq L_{4,t}(\Delta+n)-L_{3,t}(\Delta) \\ &= V_{t+1}(\Delta+n,1)-V_{t+1}(\Delta,1). \end{aligned}$$

Therefore, case 6 meets (50).

7) *Case 7:* If $L_{1,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{1,t}(\Delta+n) \leq L_{4,t}(\Delta+n) \leq L_{3,t}(\Delta+n) \leq L_{2,t}(\Delta+n)$, we can get

$$\begin{aligned} V_{t+1}(\Delta+n,0)-V_{t+1}(\Delta,0) &= L_{1,t}(\Delta+n)-L_{1,t}(\Delta) \\ V_{t+1}(\Delta+n,1)-V_{t+1}(\Delta,1) &= L_{4,t}(\Delta+n)-L_{4,t}(\Delta). \end{aligned}$$

According to (50), (51) holds.

8) *Case 8:* If $L_{1,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{1,t}(\Delta+n) \leq L_{4,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 7, (50) holds.

9) *Case 9:* If $L_{1,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{3,t}(\Delta+n) \leq L_{2,t}(\Delta+n)$, similar to case 7, (50) holds.

10) *Case 10:* If $L_{1,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 7, (50) holds.

11) *Case 11:* If $L_{1,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 6, (50) holds.

12) *Case 12:* If $L_{1,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{2,t}(\Delta) \leq L_{3,t}(\Delta)$ and $L_{1,t}(\Delta+n) \leq L_{4,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 7, (50) holds.

13) *Case 13:* If $L_{1,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{2,t}(\Delta) \leq L_{3,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 7, (50) holds.

14) *Case 14:* If $L_{1,t}(\Delta) \leq L_{4,t}(\Delta) \leq L_{2,t}(\Delta) \leq L_{3,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, we can get

$$\begin{aligned} V_{t+1}(\Delta+n,0)-V_{t+1}(\Delta,0) &= L_{2,t}(\Delta+n)-L_{1,t}(\Delta) \\ V_{t+1}(\Delta+n,1)-V_{t+1}(\Delta,1) &= L_{4,t}(\Delta+n)-L_{4,t}(\Delta) \\ &= L_{2,t}(\Delta+n)-L_{2,t}(\Delta). \end{aligned}$$

Therefore, (50) holds since $L_{1,t}(\Delta) \leq L_{2,t}(\Delta)$.

15) *Case 15:* If $L_{4,t}(\Delta) \leq L_{1,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{3,t}(\Delta+n) \leq L_{2,t}(\Delta+n)$, similar to case 7, (50) holds.

16) *Case 16:* If $L_{4,t}(\Delta) \leq L_{1,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 7, (50) holds.

17) *Case 17:* If $L_{4,t}(\Delta) \leq L_{1,t}(\Delta) \leq L_{3,t}(\Delta) \leq L_{2,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 14, (50) holds.

18) *Case 18:* If $L_{4,t}(\Delta) \leq L_{1,t}(\Delta) \leq L_{2,t}(\Delta) \leq L_{3,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 7, (50) holds.

19) *Case 19:* If $L_{4,t}(\Delta) \leq L_{1,t}(\Delta) \leq L_{2,t}(\Delta) \leq L_{3,t}(\Delta)$ and $L_{4,t}(\Delta+n) \leq L_{2,t}(\Delta+n) \leq L_{1,t}(\Delta+n) \leq L_{3,t}(\Delta+n)$, similar to case 14, (50) holds.

20) *Otherwise:* If $L_{4,t}(\Delta) \le L_{2,t}(\Delta) \le L_{1,t}(\Delta) \le L_{3,t}(\Delta)$ and $L_{4,t}(\Delta+n) \le L_{2,t}(\Delta+n) \le L_{1,t}(\Delta+n) \le L_{3,t}(\Delta+n)$, then

$$V_{t+1}(\Delta+n,0) - V_{t+1}(\Delta,0) = L_{2,t}(\Delta+n) - L_{2,t}(\Delta)$$
$$V_{t+1}(\Delta+n,1) - V_{t+1}(\Delta,1) = L_{4,t}(\Delta+n) - L_{4,t}(\Delta).$$

Based on (51), we can get that $V_{t+1}(\Delta+n,0) - V_{t+1}(\Delta,0) = V_{t+1}(\Delta+n,1) - V_{t+1}(\Delta,1)$. Therefore, (50) holds.

Above all, (50) holds in all cases. Then, by induction, we prove the inequality (50) for iteration $t \in \mathbb{N}$. As a consequence, (25) can be proved when $t \to \infty$.

Finally, we prove (26). Equation (25) can be rewritten as

$$V^*(\Delta+n,0) - V^*(\Delta+n,1) \ge V^*(\Delta,0) - V^*(\Delta,1). \quad (52)$$

While (24) reveals that

$$pV^*(\Delta+n,1) \ge pV^*(\Delta,1). \quad (53)$$

Then, combining (52) and (53) yields (26). ■

REFERENCES

[1] 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.

[2] X. Wang, C. Chen, J. He, S. Zhu, and X. Guan, "AoI-aware control and communication co-design for industrial IoT systems," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8464–8473, May 2021.

[3] J. P. Champati, M. H. Mamduhi, K. H. Johansson, and J. Gross, "Performance characterization using AoI in a single-loop networked control system," in *Proc. IEEE Int. Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 197–203.

[4] W. Y. B. Lim et al., "When information freshness meets service latency in federated learning: A task-aware incentive scheme for smart industries," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 457–466, Jan. 2022.

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

[6] 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.

[7] 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.

[8] 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.

[9] G. Chen, S. C. Liew, and Y. Shao, "Uncertainty-of-information scheduling: A restless multiarmed bandit framework," *IEEE Trans. Inf. Theory*, vol. 68, no. 9, pp. 6151–6173, Sep. 2022.

[10] X. Cao, L. Liu, Y. Cheng, and X. Shen, "Towards energy-efficient wireless networking in the big data era: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 303–332, 1st Quart., 2018.

[11] H. Harb and A. Makhoul, "Energy-efficient sensor data collection approach for industrial process monitoring," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 661–672, Feb. 2018.

[12] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 978–1013, 2nd Quart., 2018.

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

[14] Q. Chen and Y.-H. Zhu, "Scheduling channel access based on target wake time mechanism in 802.11ax WLANs," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1529–1543, Mar. 2021.

[15] 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.

[16] A. M. Bedewy, Y. Sun, and N. B. Shroff, "The age of information in multihop networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1248–1257, Jun. 2019.

[17] 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.

[18] 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.

[19] M. Klügel, M. H. Mamduhi, S. Hirche, and W. Kellerer, "AoI-penalty minimization for networked control systems with packet loss," in *Proc. IEEE Int. Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 189–196.

[20] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, "AoI-optimal joint sampling and updating for wireless powered communication systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 14110–14115, Nov. 2020.

[21] Z. Zhou, C. Fu, C. J. Xue, and S. Han, "Energy-constrained data freshness optimization in self-powered networked embedded systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2293–2306, Oct. 2020.

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

[23] J. Wang, X. Cao, B. Yin, and Y. Cheng, "Sleep–wake sensor scheduling for minimizing AoI-penalty in industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6404–6417, May 2022.

[24] D. Shah, A. Mehta, K. Patel, and A. Bartoszewicz, "Event-triggered discrete higher-order SMC for networked control system having network irregularities," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 6837–6847, Nov. 2020.

[25] M. H. Mamduhi, J. P. Champati, J. Gross, and K. H. Johansson, "Where freshness matters in the control loop: Mixed age-of-information and event-based co-design for multi-loop networked control systems," *J. Sens. Actuat. Netw.*, vol. 9, no. 3, p. 43, Sep. 2020.

[26] S. M. Ross, *Stochastic Processes*. New York, NY, USA: Wiley, 1996.

[27] A. G. Ruzzelli, P. Cotan, G. M. R. O'Hare, R. Tynan, and P. J. M. Havinga, "Protocol assessment issues in low duty cycle sensor networks: The switching energy," in *Proc. IEEE Int. Conf. Sens. Netw. Ubiquitous Trustworthy Comput. (SUTC)*, vol. 1, Jun. 2006, pp. 136–143.

[28] T. Sun and X.-M. Sun, "An adaptive dynamic programming scheme for nonlinear optimal control with unknown dynamics and its application to turbofan engines," *IEEE Trans. Ind. Informat.*, vol. 17, no. 1, pp. 367–376, Jan. 2021.

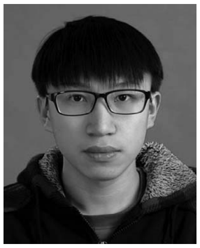
[29] A. A. Lata and M. Kang, "A review on broadcasting protocols for duty-cycled wireless sensor networks," in *Proc. Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2019, pp. 649–652.

[30] L. Li, D. Yu, Y. Xia, and H. Yang, "Remote nonlinear state estimation with stochastic event-triggered sensor schedule," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 734–745, Mar. 2019.

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.

Dr. 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.



Jia Wang (Student Member, IEEE) received the B.E. degree in control science and engineering from Harbin Engineering University, Harbin, China, in 2019, and the M.E. degree in control science and engineering from Southeast University, Nanjing, China, in 2022.

His research interests include transmission scheduling of wireless networks.

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.

Dr. Cheng was the recipient of the Best Paper Award at QShine 2007 and IEEE ICC 2011, and the Runner-Up Best Paper Award at ACM MobiHoc 2014. He was the recipient of 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 in 2016–2017. He is an Associate Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE INTERNET OF THINGS JOURNAL, and IEEE WIRELESS COMMUNICATIONS.

Jiong Jin (Member, IEEE) received the B.E. degree (with First-Class Hons.) in computer engineering from Nanyang Technological University, Singapore, in 2006, and the Ph.D. degree in electrical and electronic engineering from the University of Melbourne, Melbourne, VIC, Australia, in 2011.

From 2011 to 2013, he was a Research Fellow with the Department of Electrical and Electronic Engineering, University of Melbourne. He is currently an Associate Professor with the School of Science, Computing and Engineering Technologies, Swinburne University of Technology, Melbourne. His research interests include network design and optimization, edge computing and networking, robotics and automation, and cyber–physical systems and Internet of Things as well as their applications in smart manufacturing, smart transportation, and smart cities.