Joint Status Sampling and Updating for Minimizing Age of Information in the Internet of Things

Bo Zhou, *Member, IEEE* and Walid Saad, *Fellow, IEEE*

***Abstract*—The effective operation of time-critical Internet of things (IoT) applications requires real-time reporting of fresh status information of underlying physical processes. In this paper, a real-time IoT monitoring system is considered, in which the IoT devices sample a physical process with a sampling cost and send the status packet to a given destination with an updating cost. This joint status sampling and updating process is designed to minimize the average age of information (AoI) at the destination node under an average energy cost constraint at each device. This stochastic problem is formulated as an infinite horizon average cost constrained Markov decision process (CMDP) and transformed into an unconstrained Markov decision process (MDP) using a Lagrangian method. For the single IoT device case, the optimal policy for the CMDP is shown to be a randomized mixture of two deterministic policies for the unconstrained MDP, which is of threshold type. This reveals a fundamental tradeoff between the average AoI at the destination and the sampling and updating costs. Then, a structure-aware optimal algorithm to obtain the optimal policy of the CMDP is proposed and the impact of the wireless channel dynamics is studied while demonstrating that channels having a larger mean channel gain and less scattering can achieve better AoI performance. For the case of multiple IoT devices, a low-complexity semi-distributed suboptimal policy is proposed with the updating control at the destination and the sampling control at each IoT device. Then, an online learning algorithm is developed to obtain this policy, which can be implemented at each IoT device and requires only the local knowledge and small signaling from the destination. The proposed learning algorithm is shown to converge almost surely to the suboptimal policy. Simulation results show the structural properties of the optimal policy for the single IoT device case; and show that the proposed policy for multiple IoT devices outperforms a zero-wait baseline policy, with average AoI reductions reaching up to 33%.**

***Index Terms*—Internet of things, status update, age of information, Markov decision processes, structural analysis, distributed stochastic learning.**

I. Introduction

With the rapid proliferation of the Internet of Thing (IoT) devices, delivering timely status information of the underlying physical processes has become increasingly critical for many real-world IoT and cyber-physical system applications [2], [3], such as environment monitoring in sensor networks and vehicle tracking in smart transportation systems. Given the criticality of IoT applications, it is imperative to maintain the status information of the physical process at the destination nodes as fresh as possible, for effective monitoring and control.

This research was supported by the U.S. National Science Foundation under Grants IIS-1633363, CNS-1836802, and CNS-1460316. A preliminary version of this work has been presented at IEEE GLOBECOM 2018 [1].

B. Zhou and W. Saad are with Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA. Email: {ecebo, walids}@vt.edu.

To quantify the freshness of the status information of the physical process, the concept of *age of information* (AoI) has been proposed as a key performance metric [4] that quantifies the time elapsed since the generation of the most recent IoT device status packet received at a given destination. In contrast to conventional delay metrics, which measure the time interval between the generation and the delivery of each individual packet, the AoI considers the packet delay and the generation time of each packet, and, hence, characterizes the freshness of the status information from the perspective of the destination. Therefore, optimizing the AoI in an IoT would lead to distinctively different system designs from those used for conventional delay optimization. For example, it has been shown that the last-come-first-served (LCFS) principle achieves a lower AoI than the conventional first-come-first-served (FCFS) principle [5].

The problem of minimizing the AoI has attracted significant recent attention [4]–[17]. Generally, these works can be classified into two broad groups based on the model of the generation process of the status packets. The first group [4]–[10] models the generation process of the status packets as a queueing system in which the status packets arrive at the source node stochastically and are queued before being forwarded to the destination. Queueing theory has also been used to analyze and optimize the average AoI for FCFS [4] and LCFS systems [5]. The works in [6]–[8] propose scheduling schemes that seek to minimize the average AoI in wireless broadcast networks. In [9] and [10], the authors study the problem of AoI minimization in wireless multiaccess networks and propose decentralized scheduling policies with near-optimal performance. In the second group of works [11]–[16], the status packets can be generated at any time by the source node. The authors in [11] and [12] propose optimal updating policies to minimize the average AoI for status update systems, with a single source and multiple sources, respectively. In [13]–[15], the authors propose optimal status updating schemes for an energy harvesting source to minimize the average AoI. The authors in [16] introduce an optimal status updating scheme to minimize the average AoI under resource constraints. Motivated by recent research on AoI, the authors in [17] study the remote estimation problem for a Wiener process and propose an optimal sampling policy to minimize the estimation error.

In the existing literature, e.g., [4]–[17], the source node is usually required to perform simple monitoring tasks, such as reading a temperature sensor, and, hence, the cost for generating status packets is assumed to be negligible. However, next-generation IoT devices can now perform more complex

tasks[1], such as initial feature extraction and classification for computer vision applications, by using neural networks and on-device artificial intelligence [19], [20]. For such applications, generating the status update packets incurs energy cost for the IoT devices. Moreover, compared to the status packets generated for simple monitoring tasks (e.g., a temperature reading), a generated status packet for sophisticated artificial intelligence tasks carries richer information on the underlying physical systems (e.g., objects detected in an image or video sequence). Therefore, there will also incur some energy cost and time delay for transmitting those status packets with relatively large size to the destination node. In presence of the energy cost pertaining to the sampling and updating processes, a key open problem is to study how to intelligently sample the underlying physical systems and send status packets to the destination, in order to minimize the AoI.

The main contribution of this paper is, thus, to jointly design the status sampling and updating processes that can minimize the average AoI at the destination under an average energy cost constraint for each IoT device, by taking into account the energy cost for generating and updating status packets. In particular, our key contributions include:

- For the single IoT device case, we formulate this stochastic control problem as an infinite horizon average cost constrained Markov decision process (CMDP) and transform the CMDP into a parameterized unconstrained Markov decision process (MDP) using a Lagrangian method. We show that the optimal policy for the CMDP is a randomized mixture of two deterministic policies for the unconstrained MDP. By using the special properties of the AoI dynamics, we derive key properties of the value function for the unconstrained MDP. Based on these properties, we show that the optimal sampling and updating process for the unconstrained MDP is threshold-based with the AoI state at the device and the AoI state at the destination. This reveals a fundamental tradeoff between the average AoI at the destination and the sampling and updating costs. Then, we propose a structure-aware optimal algorithm to obtain the optimal policy for the CMDP. We also study the influence of the wireless channel fading distribution on the optimal average AoI at the destination. By using the concept of stochastic dominance, we show that channels having a larger mean channel gain and less scattering can achieve better AoI performance.
- For the case of multiple IoT devices, to obtain the optimal sampling and updating policy, we also formulate a CMDP and convert it to an unconstrained MDP. We show that the optimal sampling and updating policy, which adapts to the AoI and channels states of all IoT devices, is a function of the Q-factors of the unconstrained MDP. To overcome the curse of dimensionality and to distribute the system's controls, we propose a low-complexity semi-distributed suboptimal policy by approximating the optimal Q-factors into the sum of per-device Q-factors, based on approximate dynamic programming. Then, we propose an online learning algorithm that allows each device to learn its per-device Q-factor, which requires only the knowledge of the local AoI and channel states, as well as small signaling from the destination. The proposed semi-distributed online learning algorithm is shown to converge almost surely to the proposed suboptimal policy.
- We provide extensive simulations to illustrate additional structural properties of the optimal policy for the single device case. We show that the optimal thresholds for sampling and updating are non-decreasing with the sampling cost and the updating cost, respectively, and the optimal action is also threshold-based with respect to the channel state. For the case of multiple devices, numerical results show that the proposed semi-distributed policy outperforms a zero-wait baseline policy (i.e., sampling immediately after updating), with average AoI reductions reaching up to 33%. In summary, the derived results provide novel and holistic insights on the design of AoI-aware sampling and updating in practical IoT systems.

[1]One practical example is the Nest Cam IQ indoor security camera, which uses on-device vision processing to watch for motion, distinguish family members, and send alerts if someone is not recognized [18].

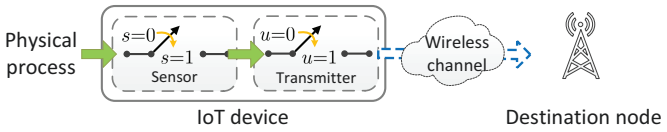


Fig. 1: Illustration of a real-time monitoring system with a single IoT device.

The rest of this paper is organized as follows. In Section II, we present the single IoT device model and analyze its properties. In Section III, we present the analysis for the case of multiple IoT devices using online learning. Section IV presents and analyzes numerical results. Finally, conclusions are drawn in Section V.

II. Optimal Sampling and Updating Control for a Single IoT Device

A. System Model

Consider a real-time IoT monitoring system composed of a single IoT device and a destination node (e.g., a base station or control center), as illustrated in Fig. 1. The IoT device encompasses a sensor which can monitor the real-time status of a physical process (referred to hereinafter as *status sampling*) and a transmitter which can send status information packets to the destination through a wireless channel (referred to hereinafter as *status updating*). For the status sampling process, different from the existing literature where the device is usually assumed to perform simple sampling tasks [4]–[17], e.g., temperature and humidity monitoring, here, we consider that the IoT device can perform more sophisticated tasks, e.g., initial feature extraction and pre-classification using machine learning and neural network tools. Hence, the time for status sampling and updating is not negligible and there will be some associated energy expenditures, which constrain the operation of the IoT device.

We consider a time-slotted system with unit slot length (without loss of generality) that is indexed by $t = 1, 2, \times\times\times$ Let $h)t[\ \emptyset\ \mathcal{H}$ be the channel state, representing the channel gain at slot t, where $\mathcal{H}$ is the finite channel state space. We assume a block fading wireless channel over all time slots and we consider an i.i.d. channel state process $\}h)t[|$ that is distributed according to a general distribution $p_{\mathcal{H}})h[$. Note that, the analytical framework and results can be readily extended to the Markovian fading channels.

1) Monitoring Model: In each slot, the IoT device must decide whether to generate a status packet and whether to send to the status packet to the destination. Let $s)t[\ \emptyset\ \}0, 1|$ be the sampling action of the device at slot t, where $s)t[= 1$ indicates that the device samples the physical process and generates a status packet at slot t, and $s)t[= 0$, otherwise. We consider that a newly generated status packet will replace the older one at the device, as the destination will not benefit from receiving an outdated status update. This is similar to the LCFS principle explored in [5]. Let C_s be the sampling cost for generating the status packet. This cost captures the computational cost needed for running some pre-classification algorithms using neural network models. We assume that the status sampling process takes one time slot. Let $u)t[\ \emptyset\ \}0, 1|$ be the update action of the device at slot t, where $u)t[= 1$ indicates that the device sends the status packet to the destination at slot t and $u)t[= 0$, otherwise. The IoT device can only send the status packet available locally. We denote by $C_u)h[$ the minimum transmission power required by the IoT device for successfully updating a status packet to the destination within a slot when the channel state is h. Without loss of generality, we assume that $C_u)h[$ is decreasing with h.

Let $\boldsymbol{w})t[\ \triangleq\ )s)t[, u)t[[\ \emptyset\ \mathcal{W} \triangleq \}0, 1| * \}0, 1|$ be the control action vector of the IoT device at t. Then, the energy cost at the device associated with action $\boldsymbol{w})t[$ is given by: $C)\boldsymbol{w})t[[\triangleq s)t[C_s + u)t[C_u)h)t[[$.

2) Age of Information Model: We adopt the AoI as the key performance metric to quantify the freshness of the status information packet [4]. The AoI is essentially defined as the time elapsed since the generation of the last status update of the physical process. Let $A_r)t[$ be the AoI at the destination at the beginning of slot t. Then, we have $A_r)t[= t\quad \delta)t[$, where $\delta)t[$ is the time slot during which the most up-to-date status packet received by the destination was generated. Note that, the IoT device can only send its currently available status packet to the destination. Thus, the AoI at the destination depends on the AoI at the device, i.e., the age of the status packet at the device. Let $A_l)t[$ be the AoI at the device at the beginning of slot t. The AoI at the device and the AoI at the destination are maintained by the device and can be implemented using counters. Let $\hat{A}_l$ and $\hat{A}_r$ be, respectively, the upper limits of the corresponding counters for the AoI at the device and the AoI at the destination. We assume that $\hat{A}_l$ and $\hat{A}_r$ are finite. This is due to that, for time-critical IoT applications, it is not meaningful for the destination node to receive a status information with an infinite age. Such highly outdated status information will not be of any use to the system or underlying application. Note that, the obtained results hold for arbitrarily finite $\hat{A}_l$ and $\hat{A}_r$, no matter how small or large $\hat{A}_l$ and $\hat{A}_r$ are. We denote by $\mathcal{A}_l \triangleq \}1, 2, \times\times\times, \hat{A}_l|$ and $\mathcal{A}_r \triangleq \}1, 2, \times\times\times, \hat{A}_r|$ the state space for the AoI at the device and the AoI at the destination, respectively. We also define $\boldsymbol{A})t[\ \triangleq\ )A_l)t[, A_r)t[[\ \emptyset\ \mathcal{A}$ as the system AoI state at the beginning of slot t, where $\mathcal{A} \triangleq \mathcal{A}_l * \mathcal{A}_r$ is the system AoI state space.

For the AoI at the device, if the device samples the physical process at slot t (i.e., $s)t[= 1$), then the AoI decreases to one (due to one slot used for status sampling), otherwise, the AoI increases by one. Then, the dynamics of the AoI at the device will be given by:

$$A_l)t + 1[= \begin{cases} 1, & \text{if } s)t[= 1, \\ \min\}A_l)t[+ 1, \hat{A}_l|, & \text{otherwise.} \end{cases} \quad (1)$$

For the AoI at the destination, if the device sends the status packet to the destination at slot t (i.e., $u)t[= 1$), then the AoI decreases to the AoI at the device at slot t plus one (due to one slot used for status packet transmission), otherwise, the AoI increases by one. Then, the dynamics of the AoI at the destination will be given by:

$$A_r)t + 1[= \begin{cases} \min\}A_l)t[+ 1, \hat{A}_r|, & \text{if } u)t[= 1, \\ \min\}A_r)t[+ 1, \hat{A}_r|, & \text{otherwise.} \end{cases} \quad (2)$$

Note that, the analytical framework can be extended to the scenario in which more than one slot are needed to generate or send a status packet, by modifying the AoI dynamics in (1) and (2), accordingly. Clearly, it may not be optimal for the device to sample the physical process immediately after updating the status. The reason is that the newly generated status packet, if not transmitted to the destination immediately (due to a possibly poor channel state), can become stale and less useful for the destination, yielding energy waste for sampling. Therefore, we are motivated to investigate how to jointly control the sampling and updating processes so as to minimize the AoI at the destination, under the stringent energy constraint at the IoT device.

B. CMDP Formulation and Optimality Equation

1) CMDP Formulation: Given an observed system AoI state $\boldsymbol{A}$ and channel state h, the IoT device determines the sampling and updating action $\boldsymbol{w}$ according to the following policy.[2]

Definition 1: A *stationary sampling and updating policy* π is defined as a mapping from the system AoI and the channel states $)\boldsymbol{A}, h[\ \emptyset\ \mathcal{A} * \mathcal{H}$ to the control action of the device $\boldsymbol{w}\ \emptyset\ \mathcal{W}$, where $\pi)\boldsymbol{A}, h[= \boldsymbol{w}$.

Under the i.i.d. assumption for the channel state process and the AoI dynamics in (1) and (2), the induced random process $\})\boldsymbol{A})t[, h)t[[|$ is a controlled Markov chain. Hereinafter, as is commonly done (e.g., see [21] and [16]), we restrict our attention to stationary unichain policies to guarantee the existence of the stationary optimal policies. For a given stationary

[2]Here, we consider the entire AoI state space of $\boldsymbol{A}$. However, in practice, one may only consider the AoI states $\boldsymbol{A}$ such that $A_r \geq A_l$ without sacrificing optimality.

unichain policy π, the average AoI at the destination and the average energy cost will be:

$$\bar{A}_r)\pi[\triangleq \limsup_{T\Rightarrow\epsilon} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\,]A_r)t[^{\cdot}, \tag{3}$$

$$\bar{C})\pi[\triangleq \limsup_{T\Rightarrow\epsilon} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\,]C)\boldsymbol{w})t[[^{\cdot}, \tag{4}$$

where the expectation is taken with respect to the measure induced by the policy π.

We seek to find the optimal sampling and updating policy that minimizes the average AoI at the destination under an average energy cost constraint at the device, as follows:

$$\bar{A}_r^{\bullet} \triangleq \min_{\pi} \bar{A}_r)\pi[, \tag{5a}$$

$$\text{s.t. } \bar{C})\pi[\geq C^{\max}. \tag{5b}$$

Here π is a stationary unichain policy and $\bar{A}_r^{\bullet}$ denotes the minimum average AoI at the destination achieved by the optimal policy $\pi^{\bullet}$ under the constraint in (5b). The problem in (5) is an infinite horizon average cost CMDP, which is to known to be challenging due to the curse of dimensionality.

2) Optimality Equation: To obtain the optimal policy $\pi^{\bullet}$ for the CMDP in (5), we reformulate the CMDP into a parameterized unconstrained MDP using the Lagrangian approach [22]. For a given Lagrange multiplier λ, we define the Lagrange cost at slot t as

$$L)\boldsymbol{A})t[, h)t[, \boldsymbol{w})t[; \lambda[\triangleq A_r)t[+ \lambda C)\boldsymbol{w})t[[. \tag{6}$$

Then, the average Lagrange cost under policy π is given by:

$$\bar{L})\pi; \lambda[\triangleq \limsup_{T\Rightarrow\epsilon} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\,]L)\boldsymbol{A})t[, h)t[, \boldsymbol{w})t[; \lambda[^{\cdot}. \tag{7}$$

Now, we have an unconstrained MDP whose goal is to minimize the average Lagrange cost:

$$\bar{L}^{\bullet})\lambda[\triangleq \min_{\pi} \bar{L})\pi; \lambda[, \tag{8}$$

where $\bar{L}^{\bullet})\lambda[$ is the minimum average Lagrange cost achieved by the optimal policy $\pi_{\lambda}^{\bullet}$ for a given λ. According to [21, Theorem 1] and [23, Theorem 4.4], we have the following relation between the optimal solutions of the problems in (5) and (8).

Lemma 1: The optimal average AoI cost in (5a) and the optimal average Lagrange cost in (8) satisfy:

$$\bar{A}_r^{\bullet} = \max_{\lambda\geq 0} \bar{L}^{\bullet})\lambda[\quad \lambda C^{\max}. \tag{9}$$

The optimal policy $\pi^{\bullet}$ of the CMDP in (5) is a randomized mixture of two deterministic stationary policies $\pi_{\lambda_1}^{\bullet}$ and $\pi_{\lambda_2}^{\bullet}$, in the form of

$$\pi^{\bullet} = \alpha\pi_{\lambda_1}^{\bullet} +)1 \quad \alpha[\pi_{\lambda_2}^{\bullet}, \tag{10}$$

where $\alpha \emptyset\ ]0, 1^{\cdot}$ is the randomization parameter, and $\pi_{\lambda_1}^{\bullet}$ and $\pi_{\lambda_2}^{\bullet}$ are the optimal policies of the unconstrained MDP in (8) under the Lagrange multipliers λ_1 and λ_2, respectively.

To obtain the optimal policy $\pi^{\bullet}$ of the CMDP, according to [24, Propositions 4.2.1, 4.2.3, and 4.2.5] (these propositions are restated in Appendix H), we first obtain the optimal policy $\pi_{\lambda}^{\bullet}$ for a given λ of the unconstrained MDP by solving the following Bellman equation.

Lemma 2: For any λ, there exists $)\theta_{\lambda}, \}V)\boldsymbol{A}, h; \lambda[|\,[$ satisfying:

$$\theta_{\lambda} + V)\boldsymbol{A}, h; \lambda[= \min_{\boldsymbol{w}\,\emptyset\,\mathcal{W}} \Big\} L)\boldsymbol{A}, h, \boldsymbol{w}; \lambda[+ \sum_{h^{\infty}\emptyset\mathcal{H}} p_{\mathcal{H}})h^{\infty}[V)\boldsymbol{A}^{\infty}, h^{\infty}, \lambda[\sum, \forall)\boldsymbol{A}, h[\ \emptyset\ \mathcal{A} * \mathcal{H}, \tag{11}$$

where $\boldsymbol{A}^{\infty}$satisfies the AoI dynamics in (1) and (2), $\theta_{\lambda} = \bar{L}^{\bullet})\lambda[$ is the optimal value to (8) for all initial state $)\boldsymbol{A})1[, h)1[[$, and $V)\}$ is the value function which is a mapping from $)\boldsymbol{A}, h[$ to real values. Moreover, for a given λ, the optimal policy achieving $\bar{L}^{\bullet})\lambda[$ will be

$$\pi_{\lambda}^{\bullet})\boldsymbol{A}, h[= \arg\min_{\boldsymbol{w}\,\emptyset\,\mathcal{W}} \Big\} L)\boldsymbol{A}, h, \boldsymbol{w}; \lambda[+ \sum_{h^{\infty}\emptyset\mathcal{H}} p_{\mathcal{H}})h^{\infty}[V)\boldsymbol{A}^{\infty}, h^{\infty}, \lambda[\sum, \forall)\boldsymbol{A}, h[\ \emptyset\ \mathcal{A} * \mathcal{H}. \tag{12}$$

From Lemma 2, we can see that $\pi_{\lambda}^{\bullet}$, which is given by (12), depends on $)\boldsymbol{A}, h[$ through the value function $V)\}$. Determining $V)\}$ involves solving the Bellman equation in (11), for which there is no closed-form solution in general [24]. Numerical algorithms such as value iteration and policy iteration are usually computationally impractical to implement for an IoT due to the curse of dimensionality and they do not typically yield many design insights. Therefore, it is desirable to analyze the structural properties of $\pi_{\lambda}^{\bullet}$, as we do next.

C. Structural Analysis and Algorithm Design

First, we characterize the structural properties of $\pi_{\lambda}^{\bullet}$ for the unconstrained MDP in (8). Then, we propose a novel structure-aware optimal algorithm to obtain the optimal policy $\pi^{\bullet}$ for the CMDP in (5). Finally, we study the effects of the wireless channel fading.

1) Optimality Properties: By using the relative value iteration algorithm and the special structures of the AoI dynamics in (1) and (2), we can prove the following property.

Lemma 3: Given $\lambda \geq 0$, $V)\boldsymbol{A}, h; \lambda[$ is non-decreasing with A_l and A_r for any $h\ \emptyset\ \mathcal{H}$.

Proof: See Appendix A. ∎

Then, we introduce the state-action Lagrange cost function, which is related to the right-hand side of the Bellman equation in (11) and is given by:

$$J)\boldsymbol{A}, h, \boldsymbol{w}; \lambda[\triangleq L)\boldsymbol{A}, h, \boldsymbol{w}; \lambda[+ \sum_{h^{\infty}\emptyset\mathcal{H}} p_{\mathcal{H}})h^{\infty}[V)\boldsymbol{A}^{\infty}, h^{\infty}, \lambda[. \tag{13}$$

We now define $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}, h; \lambda[\triangleq J)\boldsymbol{A}, h, \boldsymbol{w}; \lambda[\quad J)\boldsymbol{A}, h, \boldsymbol{w}^{\infty}, \lambda[$. If $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}, h; \lambda[\geq 0$, we say that action $\boldsymbol{w}$ *dominates* action $\boldsymbol{w}^{\infty}$at state $)\boldsymbol{A}, h[$ for a given λ. By Lemma 3, if $\boldsymbol{w}$ dominates all other actions at state $)\boldsymbol{A}, h[$ for a given λ, then we have $\pi_{\lambda}^{\bullet})\boldsymbol{A}, h[= \boldsymbol{w}$. Based on Lemma 3, we can obtain the following properties of $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}, h; \lambda[$.

Lemma 4: Given $\lambda \geq 0$, for any $\boldsymbol{A}\ \emptyset\ \mathcal{A}$, $h\ \emptyset\ \mathcal{H}$, and $\boldsymbol{w}, \boldsymbol{w}^{\infty}\emptyset\ \mathcal{W}$, $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}, h; \lambda[$ has the following properties:

A) If $\boldsymbol{w} =)0, 0[$, then $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}, h; \lambda[$ is non-decreasing with A_l for $\boldsymbol{w}^{\infty} =)1, 0[$ and non-decreasing with A_r for $\boldsymbol{w}^{\infty} =)0, 1[$ or $)1, 1[$.

B) If $\boldsymbol{w} = (0,1)$ or $(1,1)$, then $\Delta J_{\boldsymbol{w},\boldsymbol{w}^\infty}(\boldsymbol{A},h;\lambda) \geq 0$ is non-increasing with A_r for any $\boldsymbol{w}^\infty \neq \boldsymbol{w}$.

C) If $\boldsymbol{w} = (1,0)$, then $\Delta J_{\boldsymbol{w},\boldsymbol{w}^\infty}(\boldsymbol{A},h;\lambda) \geq 0$ is non-increasing with A_l for any $\boldsymbol{w}^\infty \neq \boldsymbol{w}$.

Proof: See Appendix B. ∎

Lemma 4 follows from the special properties of the AoI dynamics and is essential for the characterization of the structural properties of $\pi^\bullet_\lambda$. The property shown in Lemma 4 is similar to the diminishing-return property of multimodularity functions [25]. From Lemma 4, we can see that for a given λ and h, if action $\boldsymbol{w}$ dominates action $\boldsymbol{w}^\infty$ for some AoI state $\boldsymbol{A}$, then $\boldsymbol{w}$ still dominates $\boldsymbol{w}^\infty$ for another AoI $\boldsymbol{A}^\infty$, provided that $\boldsymbol{A}$ and $\boldsymbol{A}^\infty$ satisfy certain conditions such that $\Delta J_{\boldsymbol{w},\boldsymbol{w}^\infty}(\boldsymbol{A}^\infty,h;\lambda) \geq \Delta J_{\boldsymbol{w},\boldsymbol{w}^\infty}(\boldsymbol{A},h;\lambda) \geq 0$. Before presenting the structure of $\pi^\bullet_\lambda$ in Theorem 1, we make the following definitions:

$$\Phi_{\boldsymbol{w}}(A_r,h;\lambda) \triangleq \{A_l \in \mathcal{A}_l \text{ and } \Delta J_{\boldsymbol{w},\boldsymbol{w}^\infty}(\boldsymbol{A},h;\lambda) \geq 0 \quad \forall \boldsymbol{w}^\infty \in \mathcal{W} \text{ and } \boldsymbol{w}^\infty \neq \boldsymbol{w}\}, \tag{14}$$

$$\Psi_{\boldsymbol{w}}(A_l,h;\lambda) \triangleq \{A_r \in \mathcal{A}_r \text{ and } \Delta J_{\boldsymbol{w},\boldsymbol{w}^\infty}(\boldsymbol{A},h;\lambda) \geq 0 \quad \forall \boldsymbol{w}^\infty \in \mathcal{W} \text{ and } \boldsymbol{w}^\infty \neq \boldsymbol{w}\}. \tag{15}$$

Then, we define:

$$\phi^+_{\boldsymbol{w}}(A_r,h;\lambda) \triangleq \begin{cases} \max \Phi_{\boldsymbol{w}}(A_r,h;\lambda), & \text{if } \Phi_{\boldsymbol{w}}(A_r,h;\lambda) \neq \{\}, \\ \infty, & \text{otherwise,} \end{cases} \tag{16}$$

$$\phi_{\boldsymbol{w}}(A_r,h;\lambda) \triangleq \begin{cases} \min \Phi_{\boldsymbol{w}}(A_r,h;\lambda), & \text{if } \Phi_{\boldsymbol{w}}(A_r,h;\lambda) \neq \{\}, \\ +\infty, & \text{otherwise,} \end{cases} \tag{17}$$

$$\psi^+_{\boldsymbol{w}}(A_l,h;\lambda) \triangleq \begin{cases} \max \Psi_{\boldsymbol{w}}(A_l,h;\lambda), & \text{if } \Psi_{\boldsymbol{w}}(A_l,h;\lambda) \neq \{\}, \\ \infty, & \text{otherwise,} \end{cases} \tag{18}$$

$$\psi_{\boldsymbol{w}}(A_l,h;\lambda) \triangleq \begin{cases} \min \Psi_{\boldsymbol{w}}(A_l,h;\lambda), & \text{if } \Psi_{\boldsymbol{w}}(A_l,h;\lambda) \neq \{\}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{19}$$

Theorem 1: Given λ, for any $\boldsymbol{A} \in \mathcal{A}$ and $h \in \mathcal{H}$, there exists an optimal policy satisfying the following structural properties:

A) $\pi^\bullet_\lambda(\boldsymbol{A},h) = (0,0)$, for all $\boldsymbol{A} \in \mathcal{A}_0(h;\lambda) \triangleq \{\boldsymbol{A} \in \mathcal{A}_l \geq \phi^+_{(0,0)}(A_r,h;\lambda), A_r \geq \psi^+_{(0,0)}(A_l,h;\lambda)\}$.

B) $\pi^\bullet_\lambda(\boldsymbol{A},h) = (0,1)$ if $A_r \geq \psi_{(0,1)}(A_l,h;\lambda)$.

C) $\pi^\bullet_\lambda(\boldsymbol{A},h) = (1,0)$ if $A_l \geq \phi_{(1,0)}(A_r,h;\lambda)$.

D) $\pi^\bullet_\lambda(\boldsymbol{A},h) = (1,1)$ if $A_r \geq \psi_{(1,1)}(A_l,h;\lambda)$.

Theorem 1 characterizes the structural properties of the optimal policy $\pi^\bullet_\lambda$ of the unconstrained MDP in (8) for a given λ. Fig. 2 illustrates the analytical results of Theorem 1, where the optimal policy is computed numerically using policy iteration [26, Chapter 8.6]. Fig. 2 shows that, if the AoI state falls into the region of the black squares (i.e., $\mathcal{A}_0(h;\lambda)$), the device will remain idle and will not sample the physical process nor send the status update. Thus, $\mathcal{A}_0(h;\lambda)$ is referred to as the *idle region*. For given A_l, h, and λ, the scheduling of $(0,1)$ (or $(1,1)$) is threshold-based with respect to A_r. In other words, when A_r is small, it is not efficient to send a new status update to the destination, as a higher updating

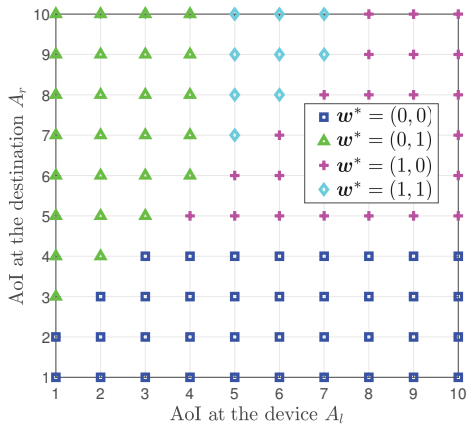


Fig. 2: Structure of the optimal policy $\pi^\bullet_\lambda$ for a given Lagrange multiplier λ and channel state h. $\hat{A}_l = \hat{A}_r = 10$. $C_s = 2$, $C_u(h) = 3.5(h$, where $h = 1,2$, $C^{\max} = 3$.

cost per age is consumed. Meanwhile, when A_r is large, it is more efficient to update the status, as the status packet at the destination becomes more outdated. For given A_r, h, and λ, the scheduling of $(1,0)$ is threshold-based with respect to A_l. Hence when A_l is small, it is not efficient to sample the physical process, as a higher sampling cost per age is incurred. In contrast, when A_l is large, it is more efficient to generate a new status packet, as the status packet at the device becomes more outdated and less useful for the destination. These observations indicate that the zero-wait policy (i.e., transmit the status packet immediately after sampling) may be detrimental to the minimization of the AoI, because of the energy cost constraint. This is reminiscent of the result in [11], however, the work in [11] obtained this outcome because of the considered random service times. *These threshold-based properties reveal a fundamental tradeoff between the AoI at the destination and the sampling and updating costs.* Such structural properties provide valuable insights for the design of the sampling and updating processes in practical IoT systems. Here, we would like to emphasize that, although the threshold-based structures may look intuitive, it is challenging to prove these structures rigorously. This is due to the coupled two AoI states and the special AoI dynamics. Moreover, it is not always possible to fully characterize the structural properties of the optimal policy, e.g., the structure with respect to h and the structures of the thresholds, as the (generally required) key property of the value function, i.e., the multimodularity [25], does not hold for our value function.

2) Algorithm Design: By exploiting the results of Theorem 1, we first propose a structure-aware algorithm to compute the optimal policy $\pi^\bullet_\lambda$ for a given λ, and, then, we describe how to update λ and obtain the optimal policy $\pi^\bullet$. Note that, although the exact values of the thresholds in Theorem 1 rely on the exact values of $V(\boldsymbol{A},h;\lambda)$, the threshold-based structure only relies on the properties of $V(\boldsymbol{A},h;\lambda)$ and $J(\boldsymbol{A},h,\boldsymbol{w};\lambda)$. These properties can be exploited to reduce the computational complexity for obtaining the optimal policy, without knowing the exact values of the thresholds. In particular, by properties B)-D) in Theorem 1, we know that the optimal action for

Algorithm 1 Structure-aware Policy Iteration Algorithm

1: Set $\pi^{\bullet}_{\lambda,0})\boldsymbol{A}, h[=)0, 0[$ for all $)\boldsymbol{A}, h[\emptyset \mathcal{A} * \mathcal{H}$, select reference state $)\boldsymbol{A}^{/}, h^{/}[$, and set $m = 0$.

2: (Policy Evaluation) Given policy $\pi^{\bullet}_{\lambda,m}$, compute the value $\theta_{\lambda,m}$ and value function $V_m)\boldsymbol{A}, h[$ from the linear system of equations[3]

$$\begin{cases} \theta_{\lambda,m} + V_m)\boldsymbol{A}, h; \lambda[= L)\boldsymbol{A}, h, \pi^{\bullet}_{\lambda,m})\boldsymbol{A}, h[; \lambda[\\ \qquad + \prod_{h^{\infty} \emptyset \mathcal{H}} p_{\mathcal{H}}) h^{\infty}[V_m)\boldsymbol{A}^{\infty}, h^{\infty}, \lambda[, \forall)\boldsymbol{A}, h[\quad , \\ V_m)\boldsymbol{A}^{/}, h^{/}; \lambda[= 0 \end{cases} \tag{21}$$

where $\boldsymbol{A}^{\infty}$satisfies the AoI dynamics in (1) and (2) under the action $\pi^{\bullet}_{\lambda,m})\boldsymbol{A}, h[$.

3: (Structured Policy Improvement) Find a new policy $\pi^{\bullet}_{\lambda,m+1}$ for each $\boldsymbol{A} \emptyset \mathcal{A}$ and $h \emptyset \mathcal{H}$, where for each $)\boldsymbol{A}, h[\emptyset \mathcal{A} * \mathcal{H}$, $\pi^{\bullet}_{\lambda,m+1})\boldsymbol{A}, h; \lambda[$ is such that:
if $\pi^{\bullet}_{\lambda,m+1}))A_l, A_r \quad 1[, h; \lambda[\quad = \quad)0, 1[,$ **then** $\pi^{\bullet}_{\lambda,m+1})\boldsymbol{A}, h; \lambda[=)0, 1[$.
else if $\pi^{\bullet}_{\lambda,m+1}))A_l \quad 1, A_r[, h; \lambda[\quad = \quad)1, 0[,$ **then** $\pi^{\bullet}_{\lambda,m+1})\boldsymbol{A}, h; \lambda[=)1, 0[$.
else if $\pi^{\bullet}_{\lambda,m+1}))A_l, A_r \quad 1[, h; \lambda[\quad = \quad)1, 1[,$ **then** $\pi^{\bullet}_{\lambda,m+1})\boldsymbol{A}, h; \lambda[=)1, 1[$.
else

$$\pi^{\bullet}_{\lambda,m+1})\boldsymbol{A}, h; \lambda[= \arg \min_{\boldsymbol{w} \emptyset \mathcal{W}} \Big\} L)\boldsymbol{A}, h, \boldsymbol{w}; \lambda[+ \sum_{h^{\infty} \emptyset \mathcal{H}} p_{\mathcal{H}}) h^{\infty}[V_m)\boldsymbol{A}^{\infty}, h^{\infty}, \lambda[\tag{22}$$

4: Go to Step 2 until $\mu^{\bullet}_{l+1} = \mu^{\bullet}_{l}$.

a certain system state is still optimal for some other system states. In particular, we can see that, for all $\boldsymbol{A}$ and h,

$$\begin{cases} \pi^{\bullet}_{\lambda})A_l, A_r, h[=)0, 1[\ \prime \ \pi^{\bullet}_{\lambda})A_l, A_r + 1, h[=)0, 1[, \\ \pi^{\bullet}_{\lambda})A_l, A_r, h[=)1, 0[\ \prime \ \pi^{\bullet}_{\lambda})A_l + 1, A_r, h[=)1, 0[, \\ \pi^{\bullet}_{\lambda})A_l, A_r, h[=)1, 1[\ \prime \ \pi^{\bullet}_{\lambda})A_l, A_r + 1, h[=)1, 1[. \end{cases} \tag{20}$$

Therefore, to find $\pi^{\bullet}_{\lambda}$, we only need to minimize in the right-hand side of (12) for some $\boldsymbol{A}$, rather than for all $\boldsymbol{A}$, which reduces the computational complexity. By incorporating (20) into a standard policy iteration algorithm, we can develop a structure-aware policy iteration algorithm, as shown in Algorithm 1. It can be seen that Algorithm 1 is a monotone policy iteration algorithm (see an example in [26, Chapter 8.11.2]), and thus, converges to the optimal policy $\pi^{\bullet}_{\lambda}$ [26, Theorem 8.6.6] (restated in Appendix H). Note that, when one of the "if" conditions in Step 3 of Algorithm 1 is satisfied for a certain system state, we can determine the optimal action immediately, without performing the minimization in (22). The computational complexity saving for each iteration in Algorithm 1 is $O)\mathfrak{M}\mathfrak{h}\mathfrak{A}\mathfrak{H}[^2[$ [27]. This is reasonable since the complexity saving grows exponentially with the state space.

From (10), we know that obtaining $\pi^{\bullet}$ requires computing the two Lagrange multipliers λ_1 and λ_2, and the randomization parameter α. As in [23] and [28], we set $\lambda_1 = \lambda^{\bullet} \quad \eta$ and $\lambda_2 = \lambda^{\bullet} + \eta$, where the perturbation parameter η is some small constant and $\lambda^{\bullet}$ is the optimal Lagrange multiplier satisfying $\lambda^{\bullet} = \min\}\lambda : \bar{C})\pi^{\bullet}_{\lambda}[\geq C^{\max}|$. By using the Robbins-Monro algorithm [29], which is a stochastic gradient-based algorithm, the Lagrange multiplier is updated according to $\lambda_{m+1} = \lambda_m + \epsilon_m \big)\bar{C})\pi^{\bullet}_{\lambda_m}[\quad C^{\max}\Big\{$, where the step $\epsilon_m = \frac{1}{m}$ and λ_1 is initialized with a sufficiently large number. The generated sequence $\}\lambda_m|$ converges to the optimal Lagrange multiplier $\lambda^{\bullet}$ [29]. Then, the randomization parameter α is given by: $\alpha =)C^{\max} \quad \bar{C})\pi^{\bullet}_{\lambda_2}[[()\bar{C})\pi^{\bullet}_{\lambda_1}[\quad \bar{C})\pi^{\bullet}_{\lambda_2}[[$. Here, α is chosen such that $\alpha\bar{C})\pi^{\bullet}_{\lambda_1}[+)1 \quad \alpha[\bar{C})\pi^{\bullet}_{\lambda_2}[= C^{\max}$. Then, for some perturbation parameter η, by [23, Theorem 4.3], we have that $\pi^{\bullet} = \alpha\pi^{\bullet}_{\lambda_1} +)1 \quad \alpha[\pi^{\bullet}_{\lambda_2}$ is the optimal policy for the CMDP. Note that, α is guaranteed to lie in $)0, 1[$ since $\bar{C})\pi^{\bullet}_{\lambda}[$ is non-increasing with λ [23]. So far, we have characterized the structural properties of the optimal policy $\pi^{\bullet}_{\lambda}$ and developed a structure-aware optimal algorithm.

3) Effects of Wireless Channel Dynamics: Next, we study the influence of the wireless channel fading distribution on the optimal average AoI at the destination. The results are established by using the stochastic dominance relations of random variables. From [21] and [30], we present the following definition.

Definition 2: Let $x)\gamma[$ be a random variable with the support on the set $\mathcal{X}$ according to a probability measure $\mu)\gamma[$ parameterized by some γ. $x)\gamma_1[$ is said to *stochastically dominate* $x)\gamma_2[$ on the set of functions $\mathcal{F}$, or $x)\gamma_1[\leftarrow_{\mathcal{F}} x)\gamma_2[$, if $\mathbb{E}]f)x)\gamma_1[[^{\cdot} \geq \mathbb{E}]f)x)\gamma_2[[^{\cdot}$, for all functions $f \emptyset \mathcal{F}$. If $\mathcal{F}$ is the set of increasing functions, then $\leftarrow_{\mathcal{F}}$ corresponds to the *first-order stochastic dominance*. If $\mathcal{F}$ is the set of increasing and concave functions, then $\leftarrow_{\mathcal{F}}$ corresponds to the *second-order stochastic dominance*.

Consider two channels I and J. Let $h^I \emptyset \mathcal{H}$ and $h^J \emptyset \mathcal{H}$ be random variables with the fading distributions $p^I_{\mathcal{H}})h[$ and $p^J_{\mathcal{H}})h[$ for channels I and J, respectively.

Theorem 2: If h^I first-order stochastically dominates h^J, then we have

$$\bar{A}^{I\bullet}_r \geq \bar{A}^{J\bullet}_r, \tag{23}$$

where $\bar{A}^{I\bullet}_r$ and $\bar{A}^{J\bullet}_r$ are the optimal average AoI at the destination under channels I and J, respectively.

Proof: See Appendix D. ∎

Theorem 2 demonstrates that channels with larger mean channel gain can achieve smaller AoI at the destination under the same resource constraint. Following the proof of Theorem 2, we have the following corollary for second-order stochastically dominating channels.

Corollary 1: If h^I second-order stochastically dominates h^J and $C_u)h[$ is decreasing and convex with h, then the optimal average AoI at the destination under channel I is smaller than that under channel J, i.e., $\bar{A}^{I\bullet}_r \geq \bar{A}^{J\bullet}_r$.

Note that, for the transmission cost $C_u)h[$ defined according to the Shannon's formula (e.g., in [21]), it can be easily seen that $C_u)h[$ satisfies the conditions of Corollary 1. If h^I has the same mean as h^J, by Definition 2, the second-order stochastic dominance of h^J over $C_u)h[$ indicates that h^I has smaller variance (i.e., less scattering) than h^J. Therefore,

[3]The solution to (21) can be derived using Gaussian elimination or the relative value iteration method [24].

Corollary 1 reveals that channels with less scattering and the same mean channel gain can achieve a smaller AoI at the destination under the same resource constraint. The results obtained in Theorem 2 and Corollary 1 reveal the fundamental monotone dependency of the optimal AoI at the destination on the transmission probability distribution of the CMDP in (5).

Thus far, we have analyzed the optimality properties for the case of a single IoT device so that to gain a deep understanding of the behavior of the optimal sampling and updating policy for the real-time monitoring system. Next, we consider a more general scenario in which there are multiple IoT devices. For such a scenario, the system state space is much larger than that for the case of a single IoT device, as it grows exponentially with the number of the devices. This hinders the structural analysis of the optimal policy and the design of an optimal algorithm with low-complexity. Therefore, we will focus on the design of a low-complexity suboptimal solution for the case of multiple IoT devices.

III. Semi-Distributed Suboptimal Sampling and Updating Control for Multiple IoT Devices

A. System Model and Problem Formulation

We now extend the real-time monitoring system in Section II to a more general scenario, in which a set $\mathcal{K}$ of K IoT devices sample the associated physical processes and update the status packets to a common destination. Hereinafter, with some notation abuse, for each IoT device $k \in \mathcal{K}$, we denote by $\boldsymbol{A}_k(t) \triangleq (A_{l,k}(t), A_{r,k}(t)) \in \mathcal{A}_k$, $h_k(t)$, and $\boldsymbol{w}_k(t) \triangleq (s_k(t), u_k(t)) \in \mathcal{W}_k$ the AoI state, the channel state, and the control action vector at slot t, respectively. Under action $\boldsymbol{w}_k(t)$, the AoI state $\boldsymbol{A}_k(t)$ for each IoT device k is updated in the same manner of (1) and (2). We define $\boldsymbol{A}(t) \triangleq (\boldsymbol{A}_k(t))_{k\in\mathcal{K}} \in \mathcal{A} \triangleq \prod_{k\in\mathcal{K}} \mathcal{A}_k$, $\boldsymbol{h}(t) \triangleq (h_k(t))_{k\in\mathcal{K}} \in \mathcal{H} \triangleq \prod_{k\in\mathcal{K}} \mathcal{H}_k$, and $\boldsymbol{w}(t) = (\boldsymbol{w}_k(t))_{k\in\mathcal{K}} \in \mathcal{W}$ as the system AoI state, the system channel state, and the system control action at slot t, respectively. Let $C_{s,k}$ and $C_{u,k}(h_k)$ be the sampling cost and the updating cost under channel state h_k of IoT device k, respectively. We assume that, the channel state processes $\{h_k(t) | k \in \mathcal{K}\}$ at the devices are mutually independent. As in [9], we consider that, in each slot, the multiple IoT devices cannot update their status packets concurrently; otherwise collisions occur and no status packets will be transmitted to the destination successfully. Thus, *different from the case of a single IoT device, the updating process of the multiple IoT devices should be carefully scheduled to avoid such collisions*. Mathematically, we have $\prod_{k\in\mathcal{K}} u_k(t) \geq 1$, for all t. Then, we define $\mathcal{W} \triangleq \mathcal{S} * \mathcal{U}$ as the feasible system control action space, where $\mathcal{S} \triangleq \{0,1\}^K$ and $\mathcal{U} \triangleq \{(u_k)_{k\in\mathcal{K}} | u_k \in \{0,1\} \forall k \in \mathcal{K}$ and $\prod_{k\in\mathcal{K}} u_k \geq 1\}$. Note that, the proposed analytical framework and algorithm design can be readily extended to support the orthogonal frequency division multiple access (OFDMA) mode, in which multiple IoT devices can update their status at the same time without collisions over different non-overlapping channels [31].

Similar to the single device case, given an observed system AoI state $\boldsymbol{A}$ and system channel state $\boldsymbol{h}$, the system control action $\boldsymbol{w}$ is derived as per the following policy.

Definition 3: A *feasible stationary sampling and updating policy* $\pi = (\pi_s, \pi_u)$ is defined as a mapping from the system AoI state and the system channel state $(\boldsymbol{A}, \boldsymbol{h}) \in \mathcal{A} * \mathcal{H}$ to the feasible system control action of the IoT devices $\boldsymbol{w} \in \mathcal{W}$, where $\pi_s(\boldsymbol{A}, \boldsymbol{h}) = \boldsymbol{s}$ and $\pi_u(\boldsymbol{A}, \boldsymbol{h}) = \boldsymbol{u}$.

Under a given stationary unichain policy π, the average AoI at the destination and the average energy cost for each IoT device k are respectively given by:

$$\bar{A}_r(\pi) \triangleq \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{k=1}^{K} A_{r,k}(t)\right], \tag{24}$$

$$\bar{C}_k(\pi) \triangleq \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} C_k(\boldsymbol{w}_k(t))\right], \forall k \in \mathcal{K}, \tag{25}$$

where $C_k(\boldsymbol{w}_k(t)) \triangleq s_k(t)(C_{s,k} + u_k(t)C_{u,k})h_k(t)$ and the expectation is taken with respect to the measure induced by the policy π.

We want to find the optimal *feasible* sampling and updating policy that minimizes the average AoI at the destination, under an average energy cost constraint for each IoT device, as follows:

$$\bar{A}_r^{\bullet} \triangleq \min_{\pi} \bar{A}_r(\pi), \tag{26a}$$

$$\text{s.t. } \bar{C}_k(\pi) \leq C_k^{\max}, \forall k \in \mathcal{K}. \tag{26b}$$

To obtain the optimal policy $\pi^{\bullet}$ in (26), we again introduce the Lagrangian for a given vector of Lagrange multipliers $\boldsymbol{\lambda} \triangleq (\lambda_k)_{k\in\mathcal{K}}$, given by:

$$\bar{L}(\pi; \boldsymbol{\lambda}) \triangleq \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} L(\boldsymbol{A}(t), \boldsymbol{h}(t), \boldsymbol{w}(t); \boldsymbol{\lambda})\right], \tag{27}$$

where $L(\boldsymbol{A}(t), \boldsymbol{h}(t), \boldsymbol{w}(t); \boldsymbol{\lambda}) \triangleq \prod_{k=1}^{K} A_{r,k}(t) + \lambda_k C_k(\boldsymbol{w}_k(t))$ is the Lagrange cost at slot t. Then, the corresponding unconstrained MDP for a given $\boldsymbol{\lambda}$ will be:

$$\bar{L}^{\bullet}(\boldsymbol{\lambda}) \triangleq \min_{\pi} \bar{L}(\pi; \boldsymbol{\lambda}), \tag{28}$$

where $\bar{L}^{\bullet}(\boldsymbol{\lambda})$ is the minimum average Lagrange cost achieved by the optimal policy $\pi_{\boldsymbol{\lambda}}^{\bullet}$ for a given $\boldsymbol{\lambda}$. The optimal average AoI at the destination in (26a) is given by $\bar{A}_r^{\bullet} = \max_{\boldsymbol{\lambda}} \bar{L}^{\bullet}(\boldsymbol{\lambda})$. In the following lemma, we summarize the solution to the unconstrained MDP in (28).

Lemma 5: For any $\boldsymbol{\lambda}$, there exists $(\theta_{\boldsymbol{\lambda}}, \{Q(\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u}; \boldsymbol{\lambda}) | \}$ satisfying:

$$\theta_{\boldsymbol{\lambda}} + Q(\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u}; \boldsymbol{\lambda}) = \min_{\boldsymbol{s}\in\mathcal{S}} \Big\{ L(\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{w}; \boldsymbol{\lambda}) + \sum_{\boldsymbol{h}^{\infty}\in\mathcal{H}} p_{\mathcal{H}}(\boldsymbol{h}^{\infty}) \min_{\boldsymbol{u}^{\infty}\in\mathcal{U}} Q(\boldsymbol{A}^{\infty}, \boldsymbol{h}^{\infty}, \boldsymbol{u}^{\infty}; \boldsymbol{\lambda}) \Big\}, \tag{29}$$

where $\boldsymbol{A}^{\infty}$ satisfies the AoI dynamics in (1) and (2) for each IoT device, $\theta_{\boldsymbol{\lambda}} = \bar{L}^{\bullet}(\boldsymbol{\lambda})$ is the optimal value to (28) for all initial state $(\boldsymbol{A}(1), \boldsymbol{h}(1), \boldsymbol{u}(1))$, and $Q(\cdot)$ is the Q-factor which is a mapping from $(\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u})$ to real values. Moreover, for a given $\boldsymbol{\lambda}$, the optimal policy achieving the optimal value $\bar{L}^{\bullet}(\boldsymbol{\lambda})$ is given by $\pi_{\boldsymbol{\lambda}}^{\bullet}(\boldsymbol{A}, \boldsymbol{h}) = (\pi_{\boldsymbol{\lambda},s}^{\bullet}(\boldsymbol{A}, \boldsymbol{h}), \pi_{\boldsymbol{\lambda},u}^{\bullet}(\boldsymbol{A}, \boldsymbol{h}))$, where $\pi_{\boldsymbol{\lambda},s}^{\bullet}(\boldsymbol{A}, \boldsymbol{h})$ attains the minimum of the right-hand side of (29) and $\pi_{\boldsymbol{\lambda},u}^{\bullet}(\boldsymbol{A}, \boldsymbol{h}) = \arg\min_{\boldsymbol{u}\in\mathcal{U}} Q(\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u}; \boldsymbol{\lambda})$.

Proof: See Appendix E. ∎

From Lemma 5, we can see that the optimal sampling and updating action depends on the Q-factor $Q(\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u}; \boldsymbol{\lambda})$ and the K Lagrange multipliers. For a given $\boldsymbol{\lambda}$, obtaining the Q-factor $Q(\cdot)$ requires solving the Bellman equation in (29), which suffers from the curse of the dimensionality due to the exponential growth of the cardinality of the system state space ($|\mathcal{A} * \mathcal{H}| = \sum_{k=1}^{K} A_{l,k}^{\max} A_{r,k}^{\max} |\mathcal{H}_k|$). Even if we could obtain the optimal Q-factors by solving (29), the derived control will be centralized thus requiring a knowledge of the system AoI states and channel states at each slot by the destination node, which is highly undesirable. Moreover, the optimal policy of the CMDP in (26) is a randomized stationary policy with a degree of randomization no greater than K [32], and, thus, may not be very suitable for practical implementations. Note that, since we need to jointly control the sampling and updating processes, our problem cannot be cast into a restless multi-armed bandit problem (RMAB) [33] as is often done in the literature[4], thus rendering the existing low-complexity solutions (e.g., [7], [8], and [9]) not applicable. Therefore, we next introduce a novel semi-distributed low-complexity algorithm to obtain a deterministic suboptimal sampling and updating policy.

B. Algorithm Design

In this subsection, we first approximate the Q-factor $Q(\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u}; \boldsymbol{\lambda})$ by the sum of the per-device Q-factor $Q_k(\boldsymbol{A}_k, h_k, u_k; \lambda_k)$. Based on the approximated Q-factor, we propose a semi-distributed sampling and updating policy, inspired by [34]. Then, we develop an online learning algorithm that enables each device to determine its per-device Q-factor and the associated Lagrange multiplier based on the observation of its AoI and channel states. Finally, we show that the proposed semi-distributed learning algorithm converges to the proposed suboptimal policy.

1) Semi-Distributed Sampling and Updating Control: To reduce the complexity for obtaining the optimal Q-factor, we adopt the linear approximation architecture [24] to approximate the Q-factor in (29) by the sum of the per-device Q-factor $Q_k(\boldsymbol{A}_k, h_k, u_k; \lambda_k)$:

$$Q(\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u}; \boldsymbol{\lambda}) \approx \sum_{k=1}^{K} Q_k(\boldsymbol{A}_k, h_k, u_k; \lambda_k), \tag{30}$$

where $Q_k(\boldsymbol{A}_k, h_k, u_k; \lambda_k)$ satisfies the following per-device Q-factor fixed point equation of each IoT device k for each given λ_k:

$$\theta_k + Q_k(\boldsymbol{A}_k, h_k, u_k; \lambda_k) = \min_{s_k \in \{0,1\}} \Big\{ L_k(\boldsymbol{A}_k, h_k, s_k, u_k; \lambda_k) + \sum_{h_k^{\infty} \in \mathcal{H}_k} p_{\mathcal{H}_k}(h_k^{\infty}) \min_{u_k^{\infty} \in \{0,1\}} Q_k(\boldsymbol{A}_k^{\infty}, h_k^{\infty}, u_k^{\infty}, \lambda_k) \Big\}, \quad \forall (\boldsymbol{A}_k, h_k, u_k) \in \mathcal{A}_k * \mathcal{H}_k * \{0,1\}. \tag{31}$$

Here, $L_k(\boldsymbol{A}_k, h_k, s_k, u_k; \lambda_k) = (A_{r,k} + \lambda_k) C_k) \boldsymbol{w}_k - C_k^{\max}$ is the per-device Lagrange cost for IoT device k. Then, according to Lemma 5, the destination node determines the updating control policy of all IoT devices based on the linear approximation in (30), given by:

$$\hat{\pi}_{\boldsymbol{\lambda},u}^{\bullet}(\boldsymbol{A}, \boldsymbol{h}) = \arg \min_{\boldsymbol{u} \in \mathcal{U}} \sum_{k=1}^{K} Q_k(\boldsymbol{A}_k, h_k, u_k; \lambda_k). \tag{32}$$

Problem (32) can be solved by a brute-force search with complexity of $O(|\mathcal{U}|)$. In particular, each IoT device k observes its AoI state $\boldsymbol{A}_k$ and channel state h_k and reports its current per-device Q-factor $\{Q_k(\boldsymbol{A}_k, h_k, u_k; \lambda_k), u_k = 0, 1\}$ to the destination node. Then, the destination node determines the system updating action $\hat{\boldsymbol{u}}^{\bullet} = \hat{\pi}_{\boldsymbol{\lambda},u}^{\bullet}(\boldsymbol{A}, \boldsymbol{h})$ according to (32) and broadcasts the updating action $\hat{\boldsymbol{u}}^{\bullet} = (\hat{u}_k^{\bullet})_{k \in \mathcal{K}}$ to the K IoT devices. Based on the local observation of $\boldsymbol{A}_k$ and h_k, as well as the updating action $\hat{u}_k^{\bullet}$, each IoT device k decides its sampling action $\hat{s}_k^{\bullet}$, which minimizes the right-hand side of (31):

$$\hat{s}_k^{\bullet} = \arg \min_{s_k \in \{0,1\}} \Big\{ L_k(\boldsymbol{A}_k, h_k, s_k, \hat{u}_k^{\bullet}; \lambda_k) + \sum_{h_k^{\infty} \in \mathcal{H}_k} p_{\mathcal{H}_k}(h_k^{\infty}) \min_{u_k^{\infty} \in \{0,1\}} Q_k(\boldsymbol{A}_k^{\infty}, h_k^{\infty}, u_k^{\infty}, \lambda_k) \Big\}. \tag{33}$$

Note that, to obtain the proposed suboptimal policy $\hat{\pi}^{\bullet}$ in (32) and (33), we need to compute $Q_k(\boldsymbol{A}_k, h_k, u_k; \lambda_k)$ by solving (31) for all IoT devices, which is a total of $O(\prod_{k=1}^{K} A_{l,k}^{\max} A_{r,k}^{\max} |\mathcal{H}_k|)$ values. However, to obtain the optimal policy $\pi^{\bullet}$, computing $Q(\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u}; \boldsymbol{\lambda})$ by solving (29) requires a total of $O(\sum_{k=1}^{K} A_{l,k}^{\max} A_{r,k}^{\max} |\mathcal{H}_k|)$ values. Therefore, the complexity of the proposed suboptimal policy decreases from exponential with K to linear with K. [5]

2) Online Stochastic Learning and Convergence Analysis: We observe that the proposed semi-distributed policy $\hat{\pi}^{\bullet}$ requires the knowledge of the per-device Q-factor and the associated Lagrange multiplier, which is challenging to obtain. Thus, we propose an online learning algorithm to estimate $Q_k(\boldsymbol{A}_k, h_k, u_k; \lambda_k)$ and λ_k at each IoT device k.

For IoT device k, based on the locally observed AoI state $\boldsymbol{A}_k(t)$, channel state $h_k(t)$, the updating action $\hat{u}_k^{\bullet}(t)$ from the destination node, and the sampling action $\hat{s}_k^{\bullet}(t)$, the per-device Q-factor and the Lagrange multiplier are respectively updated according to

$$Q_k^{t+1}(\boldsymbol{A}_k, h_k, u_k; \lambda_k^t) = Q_k^t(\boldsymbol{A}_k, h_k, u_k; \lambda_k^t) + \epsilon_{q,k}^{\upsilon_k^t(\boldsymbol{A}_k, h_k, u_k)} * \big(F_k(\boldsymbol{A}_k, h_k, u_k, \hat{s}_k^{\bullet})(t; \lambda_k^t) - F_k(\boldsymbol{A}_k^r, h_k^r, u_k^r, \hat{s}_k^{\bullet})(t^r; \lambda_k^t) + Q_k^t(\boldsymbol{A}_k, h_k, \hat{u}_k^{\bullet}; \lambda_k^t)\big), \tag{34}$$

$$\lambda_k^{t+1} = (\lambda_k^t + \epsilon_{\lambda,k}^t (C_k - \hat{\boldsymbol{w}}_k)(t))^+, \tag{35}$$

where $\mathbb{1}(\cdot)$ is the indicator function, $\upsilon_k^{\tau}(\boldsymbol{A}_k, h_k, u_k) \triangleq \sum_{\tau=1}^{t} \mathbb{1}(\boldsymbol{A}_k(\tau), h_k(\tau), \hat{u}_k^{\bullet}(\tau) = (\boldsymbol{A}_k, h_k, u_k))$ is the number of updates of the state-action pair $(\boldsymbol{A}_k, h_k, u_k)$ till t, $F_k(\boldsymbol{A}_k, h_k, u_k, s_k; \lambda_k^t) \triangleq L_k(\boldsymbol{A}_k, h_k, s_k, u_k; \lambda_k^t) +$

[4] In general, RMAB only works for the problem with only one type of control actions.

[5] The approximation error analysis of the linear approximation in (30) (which is a feature-based method) remains an open problem for CMDPs. Thus, we only provide numerical comparisons to illustrate its performance in the simulations.

Algorithm 2 Semi-Distributed Sampling and Updating Learning Algorithm.

1: **Initialization:** Set $t = 1$. Each IoT device initializes its per-device Q-factor $Q_k^t)\}$ and Lagrange multiplier λ_k^t.
2: **Updating control at the destination:** At slot t, each IoT device k reports $\}Q_k)\boldsymbol{A}_k)t[, h_k)t[, u_k; \lambda_k)t[[, u_k = 0, 1|$ to the destination node. Then, the destination node determines the system updating action according to (32) and broadcast the updating action $\hat{\boldsymbol{u}}^\bullet)t[=)\hat{u}_k^\bullet)t[[_{k\emptyset\mathcal{K}}$ to the K IoT devices.
3: **Sampling control at each IoT device:** Based on the updating action $u_k^\bullet)t[$, each IoT device k decides its sampling action $\hat{s}_k^\bullet)t[$ according to (33).
4: **Per-device Q-factor and Lagrange multiplier update at each IoT device:** Based on the current observations $\boldsymbol{A}_k)t[$ and $h_k)t[$, each IoT device k updates the per-device Q-factor $Q_k^{t+1})\boldsymbol{A}_k, h_k, u_k; \lambda_k^t[$ and λ_k^{t+1} according to (34) and (35), respectively.
5: Set $t \rightarrow t + 1$ and go to Step 2 until the convergence of $Q_k^t)\}$ and λ_k^t.

$\prod_{h_k^\infty\emptyset\mathcal{H}_k} p_{\mathcal{H}_k})h_k^\infty[\min_{u_k^\infty\emptyset\}0,1|} Q_k^t)\boldsymbol{A}_k^\infty, h_k^\infty, u_k^\infty, \lambda_k^t[$ with $\boldsymbol{A}_k^\infty$ and $\boldsymbol{A}_k$ satisfying the relations in (1) and (2), $)\boldsymbol{A}_k^r, h_k^r, u_k^r[$ is some fixed reference state-action pair, $t^r \triangleq \sup\}t\,\emptyset)\boldsymbol{A}_k)t[, h_k)t[, \hat{u}_k^\bullet)t[[=)\boldsymbol{A}_k^r, h_k^r, u_k^r[|$, and $]x^{\cdot+} = \max\}x, 0|$. $\}\epsilon_{q,k}^t|$ and $\}\epsilon_{\lambda,k}^t|$ are the sequences of step sizes satisfying:

$$\sum_t \epsilon_{q,k}^t = \infty,\ \epsilon_{q,k}^t > 0, \lim_{t\Rightarrow\infty} \epsilon_{q,k}^t = 0, \quad \sum_t \epsilon_{\lambda,k}^t = \infty,\ \epsilon_{\lambda,k}^t > 0,$$
$$\lim_{t\Rightarrow\infty} \epsilon_{\lambda,k}^t = 0, \quad \sum_t))\epsilon_{q,k}^t[^2 +)\epsilon_{\lambda,k}^t[^2[< \infty, \text{ and } \lim_{t\Rightarrow\infty} \frac{\epsilon_{\lambda,k}^t}{\epsilon_{q,k}^t} = 0. \tag{36}$$

Here, (34) is formulated following the asynchronous relative value Q-learning algorithm [35].

From (34) and (35), to implement the proposed online learning algorithm at each IoT device, we only need the local AoI and channel states, as well as the updating control action from the destination. The proposed algorithm is illustrated in Algorithm 2. It can be seen that the proposed algorithm is essentially a grant-based uplink transmission protocol (see examples in [36], [37]), which involves the exchange of messages between the IoT devices and the destination node. However, this will not incur any notable overhead, because in each slot, each IoT device needs to only transmit a few bits to exchange its per-device Q-factor value with the destination. Note that, we need to update both the per-device Q-factors and the Lagrange multipliers simultaneously. Thus, conventional value iteration and policy iteration algorithms [24], and the Q-learning algorithm under which the Lagrange multipliers are determined offline [38] are not applicable to our case.

Now, we show the almost-sure convergence of Algorithm 2. From (36), we can see that the per-device Q-factor and the Lagrange multiplier are updated concurrently, albeit over two different timescales [39]. During the update of the per-device Q-factor (timescale I), we have $\lambda_k^{t+1} \quad \lambda_k^t = O)\epsilon_{\lambda,k}^t[= o)\epsilon_{q,k}^t[$, and thus, λ_k^t can be seen as quasi-static [39] when updating $Q_k^t)\boldsymbol{A}_k, h_k, u_k; \lambda_k^t[$ in (34). We first have the following lemma on the convergence of the per-device Q-factor learning over timescale I.

Lemma 6: For step sizes $\}\epsilon_{q,k}^t|$ and $\}\epsilon_{\lambda,k}^t|$ satisfying the conditions in (36), the update of the per-device Q-factor at each IoT device k converges almost surely to the solution of the fixed point equation in(31), under any initial per-device Q-factor $Q_k^1)\}$ and the Lagrange multiplier vector $\boldsymbol{\lambda}$, i.e., $\lim_{t\Rightarrow\infty} Q_k^t)\boldsymbol{A}_k, h_k, u_k; \lambda_k[= \boldsymbol{Q}_k^\in)\boldsymbol{A}_k, h_k, u_k; \lambda_k[$, a.s. , $\forall \boldsymbol{A}_k, h_k, u_k, k$.

Proof: See Appendix F. ∎

During the update of the Lagrange multiplier (timescale II) in (35), the per-device Q-factor can be seen as nearly equilibrated [39]. Then, we have the following convergence result.

Lemma 7: The update of the vector of the Lagrange multipliers $\boldsymbol{\lambda}$ converges almost surely, i.e., $\lim_{t\Rightarrow\infty} \boldsymbol{\lambda}^t = \boldsymbol{\lambda}^\infty$ a.s., where the policy under $\boldsymbol{\lambda}^\infty$ satisfies the constraints in (26b).

Proof: See Appendix G. ∎

Based on Lemma 6 and Lemma 7, we summarize the convergence of the proposed semi-distributed online sampling and updating algorithm in Algorithm 2 in the following Theorem.

Theorem 3: For step sizes $\}\epsilon_{q,k}^t|$ and $\}\epsilon_{\lambda,k}^t|$ satisfying the conditions in (36), the iterations of the per-device Q-factor and the Lagrange multipliers in Algorithm 2 converge w.p. 1, i.e., $)\boldsymbol{Q}_k^t, \lambda_k^t[\Rightarrow)\boldsymbol{Q}_k^\in, \lambda_k^\in[$ almost surely, for each IoT device k, where $)\boldsymbol{Q}_k^\in, \lambda_k^\in[$ satisfies the fixed-point equation in (31) and the sampling and updating policy under $)\boldsymbol{Q}_k^\in, \lambda_k^\in[$ satisfies the average energy cost constraints in (26b).

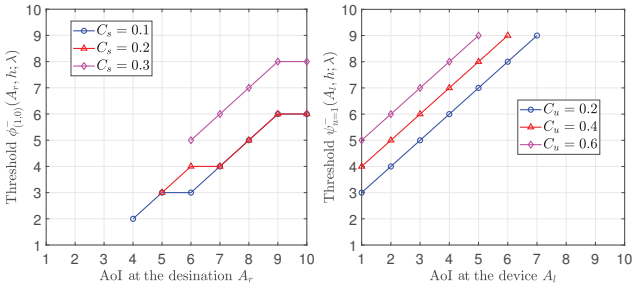
In a nutshell, we have proposed a low-complexity semi-distributed learning algorithm to find a suboptimal sampling and updating policy so as to minimize the average AoI at the destination for the case of multiple IoT devices. The proposed semi-distributed learning algorithm can be implemented at each device, requiring only the local knowledge and simple signaling from the destination, and, thus, is highly desirable for practical implementations.

IV. Simulation Results and Analysis

A. Case of A Single IoT Device

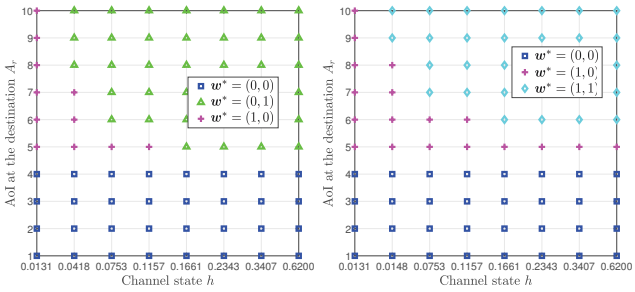
We first illustrate the structural properties of the optimal sampling and updating policy for the single IoT device case. In the simulations, we set $\mathcal{H} = \}0.0131, 0.0418, 0.0753, 0.1157, 0.1661, 0.2343, 0.3407, 0.6200|$ and the corresponding probabilities are $p_{\mathcal{H}} =]1, 1, 2, 3, 3, 2, 1, 1^{\cdot}(14$ [40]. Similar to [40], we assume that the updating cost is $C_u)h[= C_u(h$, where $C_u = 0.2$. For the sampling cost, we adopt the local-computing model in [41] and assume that $C_s = 0.2$. We set the upper limits of the AoI at the device and the AoI at the destination $\hat{A}_l$ and $\hat{A}_r$ be 10.

Fig. 3 illustrates the effects of the sampling and updating costs on the structural properties of the optimal policy $\pi_\lambda^\bullet$ for a given λ as shown in Theorem 1. In particular, Fig. 3(a) shows the relationship between the threshold $\phi_{)1,0[})A_r, h; \lambda[$



(a) (b)

Fig. 3: Impacts of the sampling and updating costs on structures of the optimal policy $\pi_\lambda^\bullet$ for a given λ in the single IoT device case. $\lambda = 0.1$. (a) Sampling cost. $h = 0.0418$. (b) Updating cost. $h = 0.1157$.



(a) (b)

Fig. 4: Structure of the optimal policy $\pi_\lambda^\bullet$ for given A_l and λ. $\lambda = 0.1$. (a) $A_l = 4$. (b) $A_l = 5$.

of choosing action $)1, 0[$ and A_r under different sampling costs C_s, for given h and λ. From Fig. 3(a), we can see, $\phi_{)1,0[}(A_r, h; \lambda[$ is non-decreasing with C_s. This indicates that the IoT device is unlikely to sample the physical process, if the sampling cost is high. Fig. 3(b) shows the relationship between $\psi_{u=1})A_l, h; \lambda[\triangleq \min\}\psi_{)0,1[})A_l, h; \lambda[, \psi_{)1,1[})A_l, h; \lambda[|$ and A_l under different updating costs C_u, for given h and λ. According to Theorem 1, if $A_r \geq \psi_{u=1})A_l, h; \lambda[$, then the optimal updating action is $u = 1$, as $\pi_\lambda^\bullet)\boldsymbol{A}, h[=)0, 1[$ or $)1, 1[$. We observe that, $\phi_{)1,0[}(A_r, h; \lambda[$ is non-decreasing with C_u. This indicates that the IoT device is not willing to send the status packet to the destination, if the updating cost is high.

Fig. 4 shows the structure of the optimal sampling and updating policy $\pi_\lambda^\bullet$ for given A_l and λ. From Fig. 4(a) and Fig. 4(b), we can see that the scheduling of action $)0, 1[$ or action $)1, 1[$ is threshold-based with respect to the channel state h. In particular, Fig. 4 shows that, if the channel state is poor, it is not efficient for the IoT device to send the status packet to the destination, as a high updating cost will be incurred. Therefore, the optimal policy can fully exploit the random nature of the wireless channel by seizing good transmission opportunities to optimize the system performance. We also notice that the optimal actions $)0, 1[$ and $)1, 1[$ do not concurrently appear in the whole state space of $)A_r, h[$, under a given A_l. This is due to the fact that the decisions of choosing $)0, 1[$ or $)1, 1[$ are threshold-based with respect to A_r and h, as seen in the upper right corners of Fig. 4(a) and Fig. 4(b).

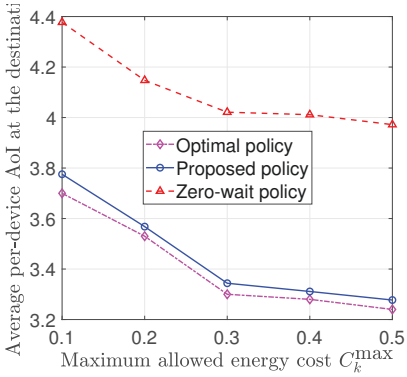
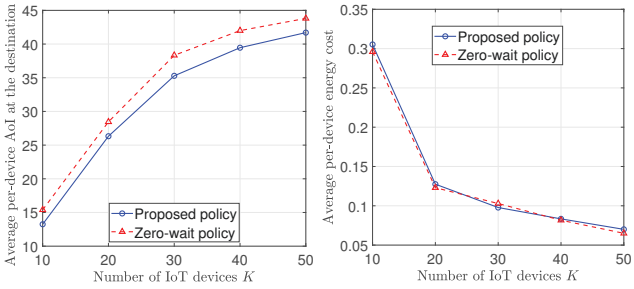


Fig. 5: Performance comparison among the optimal policy, the proposed semi-distributed policy, and the zero-wait baseline policy. $K = 2$, $A_{l,k}^{\max} = A_{r,k}^{\max} = 20, \forall k = 1, 2$.



(a) (b)

Fig. 6: Performance comparison between the proposed semi-distributed policy and the zero-wait baseline policy. $C_k^{\max} = 0.3, \forall k$. $A_{l,k}^{\max} = A_{r,k}^{\max} = 100, \forall k$. (a) Average per-device AoI at the destination. (b) Average per-device energy cost.

B. Case of Multiple IoT Devices

Next, we evaluate the performance of the proposed semi-distributed online sampling and updating policy in Algorithm 2. The system parameters are analogous to those for the single IoT device case. For each device k, the sampling cost $C_{s,k}$ is randomly selected from $]0.2, 0.3[$ and the updating cost is $C_{u,k})h_k[= C_{u,k}(h_k$, where $C_{u,k}$ is randomly selected from $]0.2, 0.3[$. We assume that the channel statistics of all IoT devices are the same, as given in Section IV-A. For comparison, consider a zero-wait baseline policy, i.e., in each slot, if an IoT device is scheduled to update its status packet, then it will sample the physical process immediately, which takes one slot. For the zero-wait baseline policy, the updating control and the updates of the per-device Q-factors and the Lagrange multipliers are similar to those of the proposed suboptimal policy, i.e., Step 2 and Step 4 in Algorithm 2. This is a commonly used baseline in the literature on AoI minimization, e.g., see [11] and references therein.

In Fig. 5, we compare the average AoI at the destination, resulting from the optimal policy, the proposed semi-randomized policy, and the zero-wait baseline policy, for two IoT devices under different values of $C_k^{\max}$. From Fig. 5, we can see that the proposed semi-distributed policy achieves a near-optimal performance and significantly outperforms the zero-wait policy.

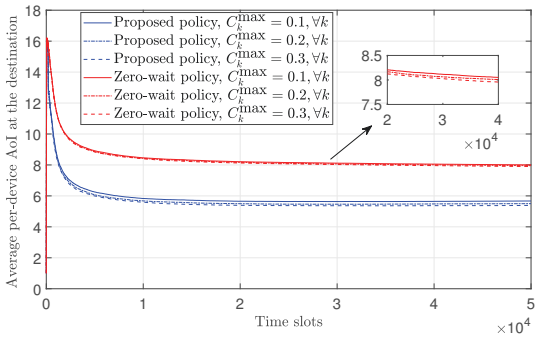


Fig. 7: Illustration of the convergence property. The number of IoT devices is $K = 5$. $A_{l,k}^{\max} = A_{r,k}^{\max} = 100, \forall k$.

Fig. 6 shows the average, per-device AoI at the destination and the average, per-device energy cost, resulting from the proposed semi-distributed policy and the zero-wait baseline policy. The simulation results are obtained by averaging over 100,000 time slots. In the simulations, for $K = 10, 20, 30, 40, 50$, it takes about 17000, 23000, 30000, 41000, 50000 time slots for the convergence of the proposed suboptimal policy, respectively. From Fig. 6, we can see that the proposed semi-distributed policy can achieve up-to 20% reduction of the average AoI at the destination over the zero-wait baseline policy, with similar energy costs. Thus the proposed policy can make better use of the limited energy at the IoT device. Moreover, for both policies, we observe that, as the number of the IoT devices increases, the average per-device AoI at the destination increases, while the average energy cost decreases. This is due to the fact that the transmission opportunities for each IoT device become lower with more IoT devices.

In Fig. 7, we show the evolution of the average per-device AoI at the destination, resulting from the proposed semi-distributed policy and the zero-wait baseline policy, under different $C_k^{\max}$. The convergence of the proposed semi-distributed learning algorithm can be clearly observed (after about 15,000 time slots). Moreover, with the increase of $C_k^{\max}$, the average per-device AoI at the destination for the two policies decreases. The performance gain of the proposed policy over the baseline policy can be as much as 33% when $C_k^{\max} = 0.3$.

V. Conclusion

In this paper, we have studied the optimal sampling and updating processes that enable IoT devices to minimize the average AoI at the destination under an average energy constraint for each IoT device in a real-time IoT monitoring system. We have formulated this problem as an infinite horizon average cost CMDP and transformed it into an unconstrained MDP. For the single IoT device case, we have shown that the optimal sampling and updating policy is of threshold type, which reveals a fundamental tradeoff between the average AoI at the destination and the sampling and updating costs. Based on this optimality property, we have proposed a structure-aware algorithm to obtain the optimal policy for the CMDP. We have also studied the effects of the wireless channel fading and shown that channels with large mean channel gain and less scattering can achieve better AoI performance. For the case of multiple IoT devices, we have shown that the optimal sampling and updating policy is a function of the Q-factors of the unconstrained MDP. To reduce the complexity in obtaining the optimal Q-factors, we have developed a semi-distributed low-complexity suboptimal policy by approximating the optimal Q-factors by a linear form of the per-device Q-factors. We have proposed an online algorithm for each device to estimate and learn its per-device Q-factor based on the locally observed AoI and channel states. We have shown the almost surely convergence of the proposed learning algorithm to the proposed suboptimal policy. Simulation results have shown that, for the single IoT device case, the optimal thresholds for sampling (updating) are non-decreasing with the sampling (updating) cost and the optimal action is threshold-based with respect to the channel state; and the proposed semi-distributed suboptimal policy for multiple IoT devices yields significant performance gain in terms of the average AoI compared to a zero-wait baseline policy. Future work will address key extensions such as providing an approximation analysis of the considered linear decomposition method and proposing grant-free uplink transmission protocols.

Appendix

A. Proof of Lemma 3

We prove Lemma 3 using the value iteration algorithm (VIA) and mathematical induction. First, we introduce the VIA [24, Chapter 4.3]. For notational convenience, we omit λ in the notation of $V)\boldsymbol{A}, h; \lambda[$. For each state $)\boldsymbol{A}, h[\emptyset \mathcal{A} * \mathcal{H}$, let $V_m)\boldsymbol{A}, h[$ be the value function at iteration m. Define the state-action cost function at iteration m as:

$$J_{m+1})\boldsymbol{A}, h, \boldsymbol{w}[\triangleq L)\boldsymbol{A}, h, \boldsymbol{w}; \lambda[+ \sum_{h^{\circ} \emptyset \mathcal{H}} p_{\mathcal{H}}) h^{\circ}[V_m)\boldsymbol{A}^{\infty}, h^{\circ}[, \quad (37)$$

where $\boldsymbol{A}^{\infty}$ is given by Lemma 2. Note that $J_{m+1})\boldsymbol{A}, h, \boldsymbol{w}[$ is related to the right-hand side of the Bellman equation in (11). For each $)\boldsymbol{A}, h[$, VIA calculates $V_{m+1})\boldsymbol{A}, h[$ according to

$$V_{m+1})\boldsymbol{A}, h[= \min_{\boldsymbol{w} \emptyset \mathcal{W}} J_{m+1})\boldsymbol{A}, h, \boldsymbol{w}[, \ \forall l. \quad (38)$$

Under any initialization of $V_0)\boldsymbol{A}, h[$, the generated sequence $\}V_m)\boldsymbol{A}, h[|$ converges to $V)\boldsymbol{A}, h[$ [24, Proposition 4.3.1], i.e.,

$$\lim_{m \Rightarrow \in} V_m)\boldsymbol{A}, h[= V)\boldsymbol{A}, h[, \ \forall)\boldsymbol{A}, h[\emptyset \mathcal{A} * \mathcal{H}, \quad (39)$$

where $V)\boldsymbol{A}, h[$ satisfies the Bellman equation in (11). Let $\pi_m^{\bullet})\boldsymbol{A}, h[$ denote the control that attains the minimum of the first term in (38) at iteration m for all $\boldsymbol{A}, h$, i.e.,

$$\pi_m^{\bullet})\boldsymbol{A}, h[= \arg \min_{\boldsymbol{w} \emptyset \mathcal{W}} J_{m+1})\boldsymbol{A}, h, \boldsymbol{w}[, \quad \forall)\boldsymbol{A}, h[\emptyset \mathcal{A} * \mathcal{H}. \quad (40)$$

We refer to $\pi_m^{\bullet} =)\pi_{s,m}^{\bullet}, \pi_{u,m}^{\bullet}[$ as the optimal policy for iteration m.

Now, consider two AoI states, $\boldsymbol{A}^1 =)A_l^1, A_r^1[$ and $\boldsymbol{A}^2 =)A_l^2, A_r^2[$. To prove Lemma 3, we only need to show that for any $\boldsymbol{A}^1, \boldsymbol{A}^2 \emptyset \mathcal{A}$, such that $A_l^2 \geq A_l^1$ and $A_r^2 \geq A_r^1$,

$$V_m)\boldsymbol{A}^2, h[\geq V_m)\boldsymbol{A}^1, h[, \quad (41)$$

holds for all $m = 0, 1, \times\times\times$

First, we initialize $V_0)\boldsymbol{A}, h[= 0$ for all $\boldsymbol{A}, h$. Thus, (41) holds for $m = 0$. Assume that (41) holds for some $m > 0$. We will prove that (41) also holds for $m + 1$. By (38), we have

$$\begin{aligned} V_{m+1})\boldsymbol{A}^1, h[&= J_{m+1}\left)\boldsymbol{A}^1, h, \pi^{\bullet}_m\right)\boldsymbol{A}^1, h[\Big\{ \\ &\overset{)a[}{\geq} J_{m+1}\left)\boldsymbol{A}^1, h, \pi^{\bullet}_m\right)\boldsymbol{A}^2, h[\Big\{ \\ &\overset{)b[}{=} A^1_r + \lambda C)\pi^{\bullet}_m)\boldsymbol{A}^2, h[[+ \sum_{h^{\circ} \emptyset \mathcal{H}} p_{\mathcal{H}})h^{\circ}[V_m)A^{1^{\infty}}_l, A^{1^{\infty}}_r, h^{\circ}[, \end{aligned} \quad (42)$$

where $)a[$ is due to the optimality of $\pi^{\bullet}_m)\boldsymbol{A}^1, h[$ for $)\boldsymbol{A}^1, h[$ in the m-th iteration, $)b[$ directly follows from (37), $A^{1^{\infty}}_l = \min\}\pi^{\bullet}_{s,m})\boldsymbol{A}^2, h[+)1 \quad \pi^{\bullet}_{s,m})\boldsymbol{A}^2, h[[)A^1_l + 1[, \hat{A}_l|$ and $A^{1^{\infty}}_r = \min\}\pi^{\bullet}_{u,m})\boldsymbol{A}^2, h[)A^1_l + 1[+)1 \quad \pi^{\bullet}_{u,m})\boldsymbol{A}^2, h[[)A^1_r + 1[, \hat{A}_r|$. By (37) and (38), we also have

$$\begin{aligned} V_{m+1})\boldsymbol{A}^2, h[&= J_{m+1}\left)\boldsymbol{A}^2, h, \pi^{\bullet}_m\right)\boldsymbol{A}^2, h[\Big\{ \\ &= A^2_r + \lambda C)\pi^{\bullet}_m)\boldsymbol{A}^2, h[[+ \sum_{h^{\circ} \emptyset \mathcal{H}} p_{\mathcal{H}})h^{\circ}[V_m)A^{2^{\infty}}_l, A^{2^{\infty}}_r, h^{\circ}[, \end{aligned} \quad (43)$$

where $A^{2^{\infty}}_l = \min\}\pi^{\bullet}_{s,m})\boldsymbol{A}^2, h[+)1 \quad \pi^{\bullet}_{s,m})\boldsymbol{A}^2, h[[)A^2_l + 1[, \hat{A}_l|$ and $A^{2^{\infty}}_r = \min\}\pi^{\bullet}_{u,m})\boldsymbol{A}^2, h[)A^2_l + 1[+)1 \quad \pi^{\bullet}_{u,m})\boldsymbol{A}^2, h[[)A^2_r + 1[, \hat{A}_r|$.

It can be seen that $A^{2^{\infty}}_l \geq A^{1^{\infty}}_l$ and $A^{2^{\infty}}_r \geq A^{1^{\infty}}_r$ for all possible $\pi^{\bullet}_m)\boldsymbol{A}^2, h[\emptyset \mathcal{W}$, which implies that $V_m)A^{2^{\infty}}_l, A^{2^{\infty}}_r, h^{\circ}[\geq V_m)A^{1^{\infty}}_l, A^{1^{\infty}}_r, h^{\circ}[$ by induction. Thus, we have $V_{m+1})\boldsymbol{A}^2, h[\geq V_{m+1})\boldsymbol{A}^1, h[$, i.e., (41) holds for $m+1$. Therefore, by induction, we can show that (41) holds for any m. By taking limits on both sides of (41) and by (39), we complete the proof of Lemma 3.

B. Proof of Lemma 4

First, we derive the general relation between $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^1, h[$ and $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^2, h[$ for any $\boldsymbol{w}, \boldsymbol{w}^{\infty} \emptyset \mathcal{W}$, $h \emptyset \mathcal{H}$, and $\boldsymbol{A}^1, \boldsymbol{A}^2 \emptyset \mathcal{A}$. Here, λ is also omitted in the notation of $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^1, h; \lambda[$ for notational convenience. By (13), we have

$$\begin{aligned} &\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^1, h[\quad \Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^2, h[\\ &= \left)L)\boldsymbol{A}^1, h, \boldsymbol{w}[+ \sum_{h^{\circ} \emptyset \mathcal{H}} p_{\mathcal{H}})h^{\circ}[V)\boldsymbol{A}^{1,\boldsymbol{w}}, h^{\circ}[\quad L)\boldsymbol{A}^1, h, \boldsymbol{w}^{\circ}[\right. \\ &\quad + \sum_{h^{\circ} \emptyset \mathcal{H}} p_{\mathcal{H}})h^{\circ}[V)\boldsymbol{A}^{1,\boldsymbol{w}^{\infty}}, h^{\circ}[\Big\{ \\ &\quad \left)L)\boldsymbol{A}^2, h, \boldsymbol{w}[+ \sum_{h^{\circ} \emptyset \mathcal{H}} p_{\mathcal{H}})h^{\circ}[V)\boldsymbol{A}^{2,\boldsymbol{w}}, h^{\circ}[\quad L)\boldsymbol{A}^2, h, \boldsymbol{w}^{\circ}[\right. \\ &\quad + \sum_{h^{\circ} \emptyset \mathcal{H}} p_{\mathcal{H}})h^{\circ}[V)\boldsymbol{A}^{2,\boldsymbol{w}^{\infty}}, h^{\circ}[\Big\{ \\ &= \sum_{h^{\circ} \emptyset \mathcal{H}} p_{\mathcal{H}})h^{\circ}[\Big)V)\boldsymbol{A}^{1,\boldsymbol{w}}, h^{\circ}[\quad V)\boldsymbol{A}^{1,\boldsymbol{w}^{\infty}}, h^{\circ}[\\ &\quad V)\boldsymbol{A}^{2,\boldsymbol{w}}, h^{\circ}[+ V)\boldsymbol{A}^{2,\boldsymbol{w}^{\infty}}, h^{\circ}[\Big\}, \end{aligned} \quad (44)$$

where $A^{1,\boldsymbol{w}}_l = \min\}s +)1 \quad s[)A^1_l + 1[, \hat{A}_l|$, $A^{1,\boldsymbol{w}}_r = \min\}u)A^1_l + 1[+)1 \quad u[)A^1_r + 1[, \hat{A}_r|$, $A^{1,\boldsymbol{w}^{\infty}}_l = \min\}s^{\infty} +)1 \quad s^{\infty}[)A^1_l + 1[, \hat{A}_l|$, $A^{1,\boldsymbol{w}^{\infty}}_r = \min\}u^{\infty})A^1_l + 1[+)1 \quad u^{\infty}[)A^1_r + 1[, \hat{A}_r|$, $A^{2,\boldsymbol{w}}_l = \min\}s +)1 \quad s[)A^2_l + 1[, \hat{A}_l|$, $A^{2,\boldsymbol{w}}_r = \min\}u)A^2_l + 1[+)1 \quad u[)A^2_r + 1[, \hat{A}_r|$, $A^{2,\boldsymbol{w}^{\infty}}_l = \min\}s^{\infty} +)1 \quad s^{\infty}[)A^2_l + 1[, \hat{A}_l|$, and $A^{2,\boldsymbol{w}^{\infty}}_r = \min\}u^{\infty})A^2_l + 1[+)1 \quad u^{\infty}[)A^2_r + 1[, \hat{A}_r|$.

Next, based on (44), we show that $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^1, h[$ is non-decreasing with A_l for $\boldsymbol{w}^{\infty} =)1, 0[$. Consider $\boldsymbol{w} =)0, 0[$, $\boldsymbol{w}^{\infty} =)1, 0[$, and $\boldsymbol{A}^1$ and $\boldsymbol{A}^2$ satisfying $A^1_l \geq A^2_1$ and $A^1_r = A^2_r$. We can see that, $A^{1,\boldsymbol{w}}_l \geq A^{2,\boldsymbol{w}}_l$, $A^{1,\boldsymbol{w}}_r = A^{2,\boldsymbol{w}}_r$, $A^{1,\boldsymbol{w}^{\infty}}_l = A^{2,\boldsymbol{w}^{\infty}}_l$, and $A^{1,\boldsymbol{w}^{\infty}}_r = A^{2,\boldsymbol{w}^{\infty}}_r$. Thus, we have $V)\boldsymbol{A}^{1,\boldsymbol{w}^{\infty}}, h^{\circ}[= V)\boldsymbol{A}^{2,\boldsymbol{w}^{\infty}}, h^{\circ}[$ and by Lemma 3, we have $V)\boldsymbol{A}^{1,\boldsymbol{w}}, h^{\circ}[\geq V)\boldsymbol{A}^{2,\boldsymbol{w}}, h^{\circ}[$. Therefore, by (44), we have $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^1, h[\quad \Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^2, h[\geq 0$, which completes the proof. Similarly, we can also prove the remaining properties of $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^1, h[$ in Lemma 4.

C. Proof of Theorem 1

We first prove Property A) of Theorem 1. Consider action $\boldsymbol{w} =)0, 0[$, action $\boldsymbol{w}^{\infty} =)1, 0[$, channel state h, AoI state $\boldsymbol{A}$ where $A_l = \phi^{+}_{)0,0[})A_r, h[$. ($\lambda$ is omitted here.) Note that, we only need to consider $\phi^{+}_{)0,0[})A_r, h[> \quad \in$. According to the definition of $\phi^{+}_{\boldsymbol{w}})A_r, h[$ in (16), we can see that $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}, h[\geq 0$, i.e., $\boldsymbol{w}$ dominates $\boldsymbol{w}^{\infty}$for state $)\boldsymbol{A}, h[$. Now, consider another AoI state $\boldsymbol{A}^{\infty}$where $A^{\infty}_l \geq A_l$ and $A^{\infty}_r = A_r$. By Lemma 4, we obtain that

$$\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^{\infty}, h[\geq \Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}, h[\geq 0, \quad (45)$$

i.e., $\boldsymbol{w} =)0, 0[$ also dominates $\boldsymbol{w}^{\infty} =)1, 0[$ for state $)\boldsymbol{A}^{\infty}, h[$. Now, we consider $\boldsymbol{w}^{\infty} =)0, 1[$ or $)1, 1[$, channel state h, AoI state $\boldsymbol{A}$ where $A_l = \psi^{+}_{)0,0[})A_l, h[$, AoI state $\boldsymbol{A}^{\infty}$where $A^{\infty}_l = A_l$ and $A^{\infty}_r \geq A_r$. According to the definition of $\psi^{+}_{\boldsymbol{w}})A_l, h[$ in (18) and Lemma 4, we can prove that (45) still holds, i.e., $\boldsymbol{w} =)0, 0[$ also dominates $\boldsymbol{w}^{\infty} =)1, 0[$ or $)1, 1[$ for state $)\boldsymbol{A}^{\infty}, h[$. By the definition of $\mathcal{A}_0)h[$, we can see that if $\boldsymbol{A} \emptyset \mathcal{A}_0)h[$, then $\boldsymbol{w} =)0, 0[$ dominates all other actions, i.e., $\pi^{\bullet})\boldsymbol{A}, h[=)0, 0[$. We complete the proof of Property A).

Next, we prove Property B) of Theorem 1. Consider action $\boldsymbol{w} =)0, 1[$, channel state h, AoI state $\boldsymbol{A}$ where $A_r = \psi_{)0,1[})A_l, h[$. We only need consider that $\psi_{)0,1[})A_l, h[< +\in$. By the definition of $\psi_{)0,1[})A_l, h[$ in (19), we have $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}, h[\geq 0$ for all $\boldsymbol{w}^{\infty} \neq \boldsymbol{w}$, i.e., $\pi^{\bullet})\boldsymbol{A}, h[=)0, 1[$. Now consider another AoI state $\boldsymbol{A}^{\infty}$where $A^{\infty}_l = A_l$ and $A^{\infty}_r \geq A_r$. By Lemma 4, we can see that $\Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}^{\infty}, h[\geq \Delta J_{\boldsymbol{w},\boldsymbol{w}^{\infty}})\boldsymbol{A}, h[\geq 0$ holds for all $\boldsymbol{w}^{\infty} \neq \boldsymbol{w}$, i.e., $\pi^{\bullet})\boldsymbol{A}^{\infty}, h[=)0, 1[$. We complete the proof of Property B). Following the proof of Property B), we can also prove Properties C) and D). This completes the proof of Theorem 1.

D. Proof of Theorem 2

To prove Theorem 2, we first prove that for any channels I and J such that h^I first-order stochastically dominates h^J,

$$V^I)\boldsymbol{A}, h; \lambda[\geq V^J)\boldsymbol{A}, h; \lambda[, \quad (46)$$

holds for all $)\boldsymbol{A}, h[$, where $V^I)\boldsymbol{A}, h; \lambda[$ and $V^J)\boldsymbol{A}, h; \lambda[$ are the value functions under channels I and J, respectively. We prove (46) through mathematical induction and the VIA in Appendix A. Similar to Appendix A, we introduce $V^I_m)\boldsymbol{A}, h; \lambda[, V^J_m)\boldsymbol{A}, h; \lambda[$, $J^I_m)\boldsymbol{A}, h, \boldsymbol{w}[$, $J^J_m)\boldsymbol{A}, h, \boldsymbol{w}[$, $\pi^{I\bullet}_m =)\pi^{I\bullet}_{s,m}, \pi^{I\bullet}_{u,m}[$, and $\pi^{J\bullet}_m =)\pi^{J\bullet}_{s,m}, \pi^{J\bullet}_{u,m}[$ for channels I and J. Since $C_u)h[$ is non-increasing with h, it can be easily shown that $V^I_m)\boldsymbol{A}, h; \lambda[$ and $V^J_m)\boldsymbol{A}, h; \lambda[$ are non-increasing with h, by

using induction and the VIA. To show (46), by (39), we only need to show that

$$V_m^I)\boldsymbol{A}, h; \lambda[\geq V_m^J)\boldsymbol{A}, h; \lambda[, \tag{47}$$

holds for $m = 0, 1, \ldots$ We initialize $V_0^I)\boldsymbol{A}, h; \lambda[= V_0^J)\boldsymbol{A}, h; \lambda[= 0$, for all $)\boldsymbol{A}, h[$, i.e., (47) holds for $m = 0$. Assume that (47) holds for some $m > 0$. We will show that (47) also holds for $m + 1$. By (37) and (38), we have

$$\begin{aligned}
&V_{m+1}^I)\boldsymbol{A}, h; \lambda[= J_{m+1}^I \Big)\boldsymbol{A}, h, \pi_m^{I\bullet})\boldsymbol{A}, h[; \lambda\Big\{ \\
&\overset{)c[}{\geq} J_{m+1}^I \Big)\boldsymbol{A}, h, \pi_m^{J\bullet})\boldsymbol{A}, h[; \lambda\Big\{ \\
&= L)\boldsymbol{A}, h, \pi_m^{J\bullet})\boldsymbol{A}, h[; \lambda[+ \sum_{h^\circ \emptyset \mathcal{H}} p_{\mathcal{H}}^I)h^\circ[V_m^I)\boldsymbol{A}^\infty, h^\circ, \lambda[\\
&\overset{)d[}{\geq} L)\boldsymbol{A}, h, \pi_m^{J\bullet})\boldsymbol{A}, h[; \lambda[+ \sum_{h^\circ \emptyset \mathcal{H}} p_{\mathcal{H}}^J)h^\circ[V_m^I)\boldsymbol{A}^\infty, h^\circ, \lambda[\\
&\overset{)e[}{\geq} L)\boldsymbol{A}, h, \pi_m^{J\bullet})\boldsymbol{A}, h[; \lambda[+ \sum_{h^\circ \emptyset \mathcal{H}} p_{\mathcal{H}}^J)h^\circ[V_m^J)\boldsymbol{A}^\infty, h^\circ, \lambda[\\
&= V_{m+1}^J)\boldsymbol{A}, h; \lambda[,
\end{aligned} \tag{48}$$

where $)c[$ is due to the optimality of $\pi_m^{I\bullet})\boldsymbol{A}, h[$ for $)\boldsymbol{A}, h[$ under channel I in the m-th iteration, $)d[$ is due to that h^I first-order stochastically dominates h^J and $V_m^I)\boldsymbol{A}, h; \lambda[$ is non-increasing with h, $)e[$ follows from the induction hypothesis $V_m^I)\boldsymbol{A}, h; \lambda[\geq V_m^J)\boldsymbol{A}, h; \lambda[$, $A_l^\infty = \min\}\pi_{s,m}^{J\bullet})\boldsymbol{A}, h[+)1 \quad \pi_{s,m}^{J\bullet})\boldsymbol{A}, h[])A_l + 1[, \hat{A}_l|$, and $A_r^\infty = \min\}\pi_{u,m}^{J\bullet})\boldsymbol{A}, h[)A_l + 1[+)1 \quad \pi_{u,m}^{J\bullet})\boldsymbol{A}, h[])A_r + 1[, \hat{A}_r|$. Thus, we prove (47) holds for $m + 1$. Then, by induction and (39), we can show that (46) holds. Based on (46), Lemma 2 and Propositions 4.3.1 in [24], we have $\bar{L}^{I\bullet})\lambda[= \theta_\lambda^I \geq \theta_\lambda^J = \bar{L}^{J\bullet})\lambda[$. Finally, by Lemma 1, we can see that $\bar{A}_r^{I\bullet} = \max_{\lambda \geq 0} \bar{L}^{I\bullet})\lambda[\quad \lambda C^{\max} \geq \max_{\lambda \geq 0} \bar{L}^{J\bullet})\lambda[\quad \lambda C^{\max} = \bar{A}_r^{J\bullet}$, which completes the proof of Theorem 2.

E. Proof of Lemma 5

For a given $\boldsymbol{\lambda}$, by Propositions 4.2.1, 4.2.3, and 4.2.5 in [24] (see Propositions 4.2.3 and 4.2.5 in Appendix H), the optimal average Lagrange cost for the unconstrained MDP in (28) is the same for all initial states and the optimal policy can be obtained by solving the following Bellman equation with respect to $)\theta_{\boldsymbol{\lambda}}, \}V)\boldsymbol{A}, \boldsymbol{h}; \boldsymbol{\lambda}[|[$.

$$\begin{aligned}
\theta_{\boldsymbol{\lambda}} + V)\boldsymbol{A}, \boldsymbol{h}; \boldsymbol{\lambda}[= \min_{\boldsymbol{w} \emptyset \mathcal{W}} \Big\{ & L)\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{w}; \boldsymbol{\lambda}[\\
& + \sum_{\boldsymbol{h}^\circ \emptyset \mathcal{H}} p_{\mathcal{H}})\boldsymbol{h}^\circ[V)\boldsymbol{A}^\infty, \boldsymbol{h}^\circ, \boldsymbol{\lambda}[\Big(\ , \ \forall)\boldsymbol{A}, \boldsymbol{h}[\emptyset \mathcal{A} * \mathcal{H},
\end{aligned} \tag{49}$$

where $V)\boldsymbol{A}, \boldsymbol{h}; \boldsymbol{\lambda}[$ is the value function. Since $\pi_{\boldsymbol{\lambda}}^\bullet)\boldsymbol{A}, \boldsymbol{h}[=)\pi_{\boldsymbol{\lambda},s}^\bullet)\boldsymbol{A}, \boldsymbol{h}[, \pi_{\boldsymbol{\lambda},u}^\bullet)\boldsymbol{A}, \boldsymbol{h}[[$, we introduce the Q-factor of state $)\boldsymbol{A}, \boldsymbol{h}[$ under updating action $\boldsymbol{u}$ as:

$$\begin{aligned}
Q)\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u}; \boldsymbol{\lambda}[\triangleq \min_{\boldsymbol{s} \emptyset \mathcal{S}} \Big\{ & L)\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{w}; \boldsymbol{\lambda}[+ \sum_{\boldsymbol{h}^\circ \emptyset \mathcal{H}} p_{\mathcal{H}})\boldsymbol{h}^\circ[V)\boldsymbol{A}^\infty, \boldsymbol{h}^\circ, \boldsymbol{\lambda}[\\
& \theta_{\boldsymbol{\lambda}}.
\end{aligned} \tag{50}$$

Thus, we have $V)\boldsymbol{A}, \boldsymbol{h}; \boldsymbol{\lambda}[= \min_{\boldsymbol{u} \emptyset \mathcal{U}} Q)\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u}; \boldsymbol{\lambda}[$ for all $)\boldsymbol{A}, \boldsymbol{h}[$ and $)\theta_{\boldsymbol{\lambda}}, \}Q)\boldsymbol{A}, \boldsymbol{h}, \boldsymbol{u}; \boldsymbol{\lambda}[|[$ satisfies the Bellman equation in (29). We complete the proof.

F. Proof of Lemma 6

Under a unichain policy defined in Definition 3, the induced random process $\})\boldsymbol{A})t[, \boldsymbol{h})t[[|$ is a controlled Markov chain with a single recurrent class and possibly some transient states [24]. According to the explanation for the condition of Proposition 4.3.2 in [24], the condition of Lemma 2 in [34] is satisfied for our problem. Then, by following the proofs of Lemma 2 in [34] and Proposition 4.3.2 in [24], we can prove the Lemma 6. The detailed proof is omitted due to page limitations.

G. Proof of Lemma 7

Due to the separation of the two timescales of the updates in (34) and (35), the update of the Q-factors can be regarded as converged to $\boldsymbol{Q}_k^\in)\boldsymbol{\lambda}^t[$ under $\boldsymbol{\lambda}^t$ [39]. Then, by the theory of stochastic approximation [34], [39], [42], the iterations of the update of the Lagrange multiplier in (35) can be described by the following Ordinary Differential Equation (ODE):

$$\boldsymbol{\lambda}^t = \mathbb{E}^{\pi_{\boldsymbol{\lambda}^t}^\bullet}]C_1)\hat{\boldsymbol{w}}_1)t[[\quad C_1^{\max}, \ldots, C_K)\hat{\boldsymbol{w}}_K)t[[\quad C_K^{\max}\cdot, \tag{51}$$

where $\pi_{\boldsymbol{\lambda}^t}^\bullet$ is the converged control policy in Algorithm 2 under $\boldsymbol{\lambda}^t$ and the expectation is taken with respect to the measure induced by the policy $\pi_{\boldsymbol{\lambda}^t}^\bullet$. Denote $\bar{L})\boldsymbol{\lambda}^t[= \mathbb{E}^{\pi_{\boldsymbol{\lambda}^t}^\bullet}]\prod_{k=1}^K)A_{r,k} + \lambda_k)C_k)\hat{\boldsymbol{w}}_k[\quad C_k^{\max}[[\cdot$. Since the sampling and updating actions are discrete, we have $\pi_{\boldsymbol{\lambda}^t}^\bullet = \pi_{\boldsymbol{\lambda}^t + \boldsymbol{\delta}_\lambda}^\bullet$. By chain rule, it can be seen that $\frac{\partial \bar{L})\boldsymbol{\lambda}^t[}{\partial \lambda_k^t} = \mathbb{E}^{\pi_{\boldsymbol{\lambda}^t}^\bullet}]C_k)\hat{\boldsymbol{w}}_k)t[[\quad C_k^{\max}\cdot$. Thus, the ODE in (51) can be expressed as $\boldsymbol{\lambda}^t = \nabla \bar{L})\boldsymbol{\lambda}^t[$. Therefore, the ODE in (51) will converge to $\arg\max \bar{L})\boldsymbol{\lambda}^\in[$, which corresponds to $\nabla \bar{L})\boldsymbol{\lambda}^\in[= 0$. In other words, the policy under $)\boldsymbol{Q}^\in, \boldsymbol{\lambda}^\in[$ satisfies the constraint in (26b). This completes the proof.

H. Some preliminaries on MDP

Proposition 4.2.3 in [24]: Let the weak accessibility (WA) condition hold. Then the optimal average cost is the same for all initial states.

Proposition 4.2.5 in [24]: If all stationary policies are unichain, the WA condition holds.

Theorem 8.6.6 in [26]: Suppose all stationary policies are unichain, and the set of states and actions are finite, then policy iteration converges in a finite number of iterations to the optimal policy satisfying the Bellman equation.

REFERENCES

[1] B. Zhou and W. Saad, "Optimal sampling and updating for minimizing age of information in the Internet of Things," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010.

[3] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3949–3963, June 2016.

[4] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*, Orlando, FL, USA, March 2012, pp. 2731–2735.

[5] S. Kaul, R. D. Yates, and M. Gruteser, "Status updates through queues," in *Proc. of 46th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, USA, March 2012, pp. 1–6.

[6] Y.-P. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *arXiv preprint arXiv:1712.07419*, 2017.

[7] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *arXiv preprint arXiv:1801.01803*, 2018.

[8] Y.-P. Hsu, "Age of information: Whittle index for scheduling stochastic arrivals," in *Proc. of IEEE International Symposium on Information Theory (ISIT)*, Colorado, USA, June 2018.

[9] Z. Jiang, B. Krishnamachari, S. Zhou, and Z. Niu, "Can decentralized status update achieve universally near-optimal age-of-information in wireless multiaccess channels?" in *Proc. of IEEE The International Teletraffic Congress (ITC)*, Vienna, Austria, Sep. 2018.

[10] Z. Jiang, B. Krishnamachari, X. Zheng, S. Zhou, and Z. Niu, "Timely status update in massive IoT systems: Decentralized scheduling for wireless uplinks," *arXiv preprint arXiv:1801.03975*, 2018.

[11] 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.

[12] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, "Age-optimal sampling and transmission scheduling in multi-source systems," *arXiv preprint arXiv:1812.09463*, 2018.

[13] X. Wu, J. Yang, and J. Wu, "Optimal status update for age of information minimization with an energy harvesting source," *IEEE Trans. Green Commun. and Netw.*, vol. 2, no. 1, pp. 193–204, March 2018.

[14] B. T. Bacinoglu and E. Uysal-Biyikoglu, "Scheduling status updates to minimize age of information with an energy harvesting sensor," *arXiv preprint arXiv:1701.08354*, 2017.

[15] S. Feng and J. Yang, "Minimizing age of information for an energy harvesting source with updating failures," in *Proc. of IEEE International Symposium on Information Theory (ISIT)*, Colorado, USA, June 2018, pp. 2431–2435.

[16] E. T. Ceran, D. Gunduz, and A. Gyorgy, "Average age of information with hybrid ARQ under a resource constraint," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, Spain, April 2018.

[17] Y. Sun, Y. Polyanskiy, and E. Uysal-Biyikoglu, "Remote estimation of the wiener process over a channel with random delay," *arXiv preprint arXiv:1701.06734*, 2017.

[18] Nest Cam IQ indoor security camera, https://nest.com/cameras/nest-cam-iq-indoor/overview/.

[19] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *arXiv preprint arXiv:1710.02913*, 2017.

[20] S. Teerapittayanon, B. McDanel, and H. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *Proc. of IEEE International Conference on Distributed Computing Systems (ICDCS)*, Atlanta, GA, USA, June 2017, pp. 328–339.

[21] D. V. Djonin and V. Krishnamurthy, "MIMO transmission control in fading channels–a constrained Markov decision process formulation with monotone randomized policies," *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5069–5083, 2007.

[22] E. Altman, *Constrained Markov Decision Processes*. London, U.K.: Chapman & Hall, 1999.

[23] F. J. Beutler and K. W. Ross, "Optimal policies for controlled Markov chains with a constraint," *Journal of Mathematical Analysis and Applications*, vol. 112, no. 1, pp. 236 – 252, 1985.

[24] D. P. Bertsekas, *Dynamic programming and optimal control, 3rd edition, volume II*. Belmont, MA: Athena Scientific, 2011.

[25] G. Koole, "Monotonicity in Markov reward and decision chains: Theory and applications," *Foundations and Trends in Stochastic Systems*, vol. 1, no. 1, pp. 1–76, 2006.

[26] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. New York, NY, USA: Wiley, 2009, vol. 414.

[27] M. L. Littman, T. L. Dean, and L. P. Kaelbling, "On the complexity of solving markov decision problems," in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, 1995, pp. 394–402.

[28] M. H. Ngo and V. Krishnamurthy, "Monotonicity of constrained optimal transmission policies in correlated fading channels with ARQ," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 438–451, Jan 2010.

[29] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 09 1951.

[30] J. E. Smith and K. F. McCardle, "Structural properties of stochastic dynamic programs," *Operations Research*, vol. 50, no. 5, pp. 796–809, 2002.

[31] A. D. Zayas and P. Merino, "The 3GPP NB-IoT system architecture for the Internet of Things," in *Proc. of IEEE International Conference on Communications Workshops (ICC Workshops)*, Paris, France, May 2017, pp. 277–282.

[32] K. W. Ross, "Randomized and past-dependent policies for Markov decision processes with multiple constraints," *Operations Research*, vol. 37, no. 3, pp. 474–477, 1989.

[33] J. Gittins, K. Glazebrook, and R. Weber, *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[34] Y. Cui and V. K. N. Lau, "Distributive stochastic learning for delay-optimal OFDMA power and subband allocation," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4848–4858, Sept 2010.

[35] J. Abounadi, D. Bertsekas, and V. S. Borkar, "Learning algorithms for markov decision processes with average cost," *SIAM J. Control Optim.*, vol. 40, no. 3, pp. 681–698, 2001.

[36] Y. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3GPP narrowband Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 117–123, March 2017.

[37] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: Issues and approaches," *IEEE Commun Mag*, vol. 51, no. 6, pp. 86–93, 2013.

[38] D. V. Djonin and V. Krishnamurthy, "Q-learning algorithms for constrained markov decision processes with randomized monotone policies: Application to mimo transmission control." *IEEE Trans. Signal Processing*, vol. 55, no. 5-2, pp. 2170–2181, 2007.

[39] V. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency, 2008.

[40] K. T. Phan, T. Le-Ngoc, M. van der Schaar, and F. Fu, "Optimal scheduling over time-varying channels with traffic admission control: Structural results and online learning algorithms," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4434–4444, September 2013.

[41] C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, March 2017.

[42] V. S. Borkar, "Stochastic approximation with two time scales," *Systems & Control Letters*, vol. 29, no. 5, pp. 291 – 294, 1997.