AoI Minimization in Status Update Control With Energy Harvesting Sensors

Mohammad Hatami, Markus Leinonen, *Member, IEEE*, and Marian Codreanu, *Member, IEEE*

***Abstract*— Information freshness is crucial for time-critical IoT applications, e.g., monitoring and control. We consider an IoT status update system with users, energy harvesting sensors, and a cache-enabled edge node. The users receive time-sensitive information about physical quantities, each measured by a sensor. Users demand for the information from the edge node whose cache stores the most recently received measurements from each sensor. To serve a request, the edge node either commands the sensor to send an update or retrieves the aged measurement from the cache. We aim at finding the best actions of the edge node to minimize the average AoI of the served measurements at the users, termed on-demand AoI. We model this problem as a Markov decision process and develop reinforcement learning (RL) algorithms: model-based value iteration and model-free Q-learning. We also propose a Q-learning method for the realistic case where the edge node is informed about the sensors' battery levels only via the status updates. The case under transmission limitations is also addressed. Furthermore, properties of an optimal policy are characterized. Simulation results show that an optimal policy is a threshold-based policy and that the proposed RL methods significantly reduce the average cost compared to several baselines.**

***Index Terms*— Internet of Things (IoT), age of information (AoI), energy harvesting, reinforcement learning (RL), value iteration algorithm (VIA), dynamic programming, Q-learning.**

I. Introduction

INTERNET of Things (IoT) is an emerging technology to connect different devices to enable emergent applications with minimal human intervention. IoT enables the users to effectively interact with the physical surrounding environment and empower context-aware applications like smart cities [1]. A typical IoT network consists of multiple wireless sensors which measure physical phenomena and communicate the obtained measurements to a destination for further processing, e.g., to perform distributed target detection [2]. Two inherent features of such networks are: 1) stringent energy limitations of battery-powered sensors which, however, may be counteracted by *harvesting* energy[1] from environmental sources such as sun, heat, and RF ambient [5], [6], and 2) *transient* nature of data, i.e., the sensors' measurements become outdated after a while. This calls for the design of IoT sensing techniques where the sensors sample and send a minimal number of measurements to conserve the energy while providing the end users highly fresh data, as required by time-sensitive applications. The freshness of information can be quantified by the recently emerged metric, the *age of information* (AoI) [7]–[11]. Formally, AoI is defined as the time elapsed since the latest successfully received status update packet at the destination was generated at a source node. We introduce *on-demand AoI* that represents the AoI at the users restricted to the users' request instants. The works that address AoI in IoT networks can be divided into two main classes: 1) the works that focus on analyzing the AoI in a specific scenario under their proposed status update control/scheduling policies [12]–[16], and 2) the works that focus on finding an optimal control/scheduling policy for a specific system. For the latter class, there are two main approaches. The first approach involves finding an optimal policy by applying different tools from optimization theory [17]–[23]. Such approaches need exact information about the models and statistics of the environment, e.g., the EH probabilities of sensors. The second category includes designs relying on dynamic programming and learning methods [24]–[32]. In this paper, we focus on this category and find an optimal policy that minimizes the AoI about the sensors' measurements received by the users in an EH IoT network.

A particular interest has arisen in designing AoI-aware IoT networks [12], [13]. In [12], a threshold-based age-dependent random access algorithm was proposed for massive IoT networks, in which an IoT device sends an update when its age exceeds a predefined threshold. In [13], the authors presented a stochastic geometry analysis for the average AoI in a cellular IoT network.

Manuscript received September 9, 2020; revised March 19, 2021 and July 20, 2021; accepted September 10, 2021. Date of publication September 22, 2021; date of current version December 16, 2021. This research has been financially supported in part by Infotech Oulu, the Academy of Finland (grant 323698), and Academy of Finland 6Genesis Flagship (grant 318927). Marian Codreanu would like to acknowledge the support of the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 793402 (COMPRESS NETS). The work of Markus Leinonen has also been financially supported in part by the Academy of Finland (grant 319485). Mohammad Hatami would like to acknowledge the support of HPY Research Foundation and Riitta ja Jorma J. Takanen Foundation. This article was presented in part at the 31th Annual International Symposium on Personal, Indoor and Mobile Radio Communications, London, U.K., in 2020 [DOI: 10.1109/PIMRC48278.2020.9217302]. The associate editor coordinating the review of this article and approving it for publication was L. Cai. *(Corresponding author: Mohammad Hatami.)*

Mohammad Hatami and Markus Leinonen are with the Centre for Wireless Communications—Radio Technologies, University of Oulu, 90014 Oulu, Finland (e-mail: mohammad.hatami@oulu.fi; markus.leinonen@oulu.fi).

Marian Codreanu is with the Department of Science and Technology, Linköping University, 581 83 Linköping, Sweden (e-mail: marian.codreanu@liu.se).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2021.3114681.

Digital Object Identifier 10.1109/TCOMM.2021.3114681

[1] An alternative approach for ultra-low-power IoT sensors is ambient backscatter communications; see e.g., [3], [4].

AoI has also been investigated in cache updating systems [17], [18]. In [17], the authors introduced a popularity-weighted AoI metric for updating dynamic content in a local cache, where the content is subjected to version updates. The authors in [18] considered a cache updating system with a source, a cache, and a user, and found an analytical expression for the average freshness of the files at the user under the proposed threshold policy.

The works [14]–[16] focused on analyzing the AoI in EH IoT networks. The authors in [14] considered a known EH model and proposed a threshold adaptation algorithm to maximize the hit rate in an IoT sensing network. In [15], the authors analyzed the average AoI in a cache enabled status updating system with an EH sensor. In [16], the author derived a closed-form expression for the average AoI in a wireless powered sensor network.

Age-optimal policies for status update packet transmissions in EH networks have been derived in [19]–[23] by using different methods from optimization theory. In [19], the authors derived an optimal policy for an EH source that sends updates to a network interface queue for delivery to a monitoring system. In [20], the authors derived age-optimal online policies for an EH sensor having a unit-sized or infinite battery using renewal theory. In [21], the authors explored the benefits of erasure status feedback for online timely updating for an EH sensor with a unit-sized battery. Age-optimal transmission policies for EH two-hop networks were investigated in [22]. In [23], the authors derived age-optimal policies for an EH sensor with a finite-sized battery.

Several works have developed an AoI-optimal status update systems by using dynamic programming and learning based methods [24]–[32]. A commonality in these works is to model the problem as a Markov decision process (MDP), and find an optimal policy using model-based reinforcement learning (RL) methods based on dynamic programming, e.g., value iteration algorithm (VIA), and/or model-free RL methods, e.g., Q-learning. A comprehensive survey of RL based methods for autonomous IoT networks was presented in [33]. The authors in [24] used deep RL to solve a cache replacement problem with a limited cache size and transient data in an IoT network. Minimizing AoI in a wireless ad hoc network via deep RL was investigated in [25]. The authors of [26] derived optimal sampling and updating policies that minimize the average AoI in an IoT monitoring system. In [27], deep RL was used to minimize AoI in a multi-node monitoring system, in which the sensors are powered through wireless energy transfer by the destination. The authors of [28] derived age-optimal sampling instants for an EH sensor with known EH statistics. In [29], the authors investigated age-optimal policies where an EH sensor takes advantage of multiple available transmission modes. In [30], the authors studied AoI minimization in cognitive radio EH communications. In [31], the authors studied age-optimal policies for an EH device that monitors a stochastic process, which can be in either a normal or an alarm state of operation. In [32], the authors studied age-optimal policies for cases where the channel and EH statistics are either known or unknown.

Majority of the existing works, including all the above ones, investigate the AoI minimization in cases where the updates are relevant to the monitoring entity at all time moments. Only a few works studied a concept similar to the on-demand AoI herein. In [34], the authors introduced the idea of effective AoI (EAoI) under a generic request-response model where a server serves the users with time-sensitive information. They elaborated on the fact that minimizing the time-average EAoI is in general different from minimizing the time-average AoI. In [35], the authors studied an information-update system where a user pulls information from servers. However, in contrast to our paper, the works [34], [35] do not consider energy limitation at the source nodes.

A. Contributions

We consider an IoT status update network that consists of EH IoT sensors, a cache-enabled edge node, and the users. The users receive time-sensitive information about physical quantities, each of which is measured by a sensor. The users demand for the information from the edge node (a gateway) whose cache stores the most recently received measurements of each physical quantity. To serve a user's request, the edge node can either command the corresponding sensor to send a fresh measurement in the form of status update packet over an unreliable channel, or use the aged data in the cache. The former enables serving a user with fresh measurement, yet consuming energy from the sensor's battery. The latter prevents the activation of the sensors for every request so that the sensors can utilize the sleep mode to save a considerable amount of energy [14], but the data received by the users becomes stale. This results in an inherent *trade-off* between the AoI at the users and conservation of the sensors' energy in the finite batteries.

We aim to find the best action of the edge node at each time slot, called an optimal policy, to minimize the average AoI about the physical quantities at the users restricted to the users' request moments, i.e., average on-demand AoI. The on-demand AoI minimization is different from the conventional AoI optimization in that the freshness of information is only important when user(s) need the information. To tackle this status update control problem, we derive an MDP model and propose RL based algorithms to obtain optimal policies under different circumstances in the learning environment. To summarize, our main contributions are:

- First, we derive an MDP model for the on-demand AoI minimization problem, calculate the state transition probabilities, and propose a model-based VIA to find an optimal policy.
- Then, for the case where the state transition probabilities are unknown, we propose a model-free online Q-learning method to search for an optimal policy. As a practical consideration, we also propose an online method for the realistic scenario where the edge node is informed about the sensors' battery levels only via the status updates.
- We next derive structural properties of the optimal policy – obtained by VIA – and show that the optimal policy

has a threshold-based structure with respect to the AoI in a specific scenario.
- In addition, we investigate a massive IoT scenario where the edge node can command only a limited number of sensors. In particular, we find an optimal policy and propose a low-complexity sub-optimal algorithm.
- Extensive numerical experiments are conducted to show that an optimal policy is a threshold-based policy and that the proposed RL algorithms significantly reduce the average on-demand AoI as compared to several baseline policies.

Our paper has certain relations to [20]–[32], [34], yet with the following differences. The works [20]–[23] focus on a continuous-time single EH sensor and use optimization methods different to the MDP based learning methods herein. The works [24]–[26], [34], do not consider energy limitations at the source nodes, whereas we consider EH sensors with finite batteries. In [27], each time slot is allocated either to one sensor to send an update or to the destination to broadcast RF energy signals to charge the sensors; in our system model, all the users' requests in the network are handled by the edge node at each time slot, and the sensors harvest energy from the environment. In [28]–[32], the authors studied AoI-optimal policies for a single EH sensor that sends updates to a destination in cases where the updates are relevant to the monitoring entity at all time moments, whereas we investigate on-demand AoI minimization in IoT networks where EH sensors send updates to the users via a cache-enabled edge node. Different from all the above works, we propose a learning based approach for the case where the edge node is informed about the sensors' battery levels only via the status update packets, i.e., partial battery knowledge at the edge node. To the best of our knowledge, this is the first work that investigates on-demand AoI in an EH IoT network and proposes MDP based learning approaches for age-aware status update control with EH sensors. A comparative summary of contributions is presented in Table I. Preliminary results of this paper appear in [36].

B. Organization

The paper is organized as follows. Section II presents the system model and problem definition. A Markov decision process and definition of optimal policies are presented in Section III. Our proposed RL-based status update control algorithms are developed in Section IV. Structural properties of an optimal policy are analytically characterized in Section V. The scenario under the transmission limitation is addressed in Section VI. Simulation results are presented in Section VII. Concluding remarks are drawn in Section VIII.

C. Notations

Vectors and sets are written in boldface lower ($\mathbf{a}$) and calligraphy ($\mathcal{S}$) letters, respectively. The expectation operation is denoted as $\mathbb{E}[\cdot]$. The cardinality of a set $\mathcal{S}$ is denoted as $|\mathcal{S}|$. The indicator function $\mathbb{1}_{\{.\}}$ is equal to 1 (only) whenever the condition $\{.\}$ is true.

II. System Model and Problem Formulation

A. Network Model

We consider an IoT sensing network consisting of multiple users (*data consumers*), a wireless edge node, and a set $\mathcal{K} = \{1, \ldots, K\}$ of K energy harvesting (EH) sensors (*data producers*), as depicted in Fig. 1. Users are interested in time-sensitive information about physical quantities (e.g., temperature or humidity) which are independently measured by the K sensors; formally, sensor $k \in \mathcal{K}$ measures a physical quantity f_k. We assume that there is no direct link between the users and the sensors, and the edge node acts as a gateway between them. Thus, the users' requests for the values of f_k, $k \in \mathcal{K}$, are served (only) via the edge node.

The system operates in a slotted time fashion, i.e., time is divided into slots labeled with discrete indices $t \in \mathbb{N}$. At the beginning of slot t, users request for the values of physical quantities f_k from the edge node. Formally, let $r_k(t) \in \{0, 1\}$, $t = 1, 2, \ldots$, denote the random process of requesting the value of f_k at the beginning of slot t; $r_k(t) = 1$ if the value of f_k is requested and $r_k(t) = 0$ otherwise. Note that at each time slot, there can be multiple requests arriving at the edge node.

The edge node is equipped with a cache storage that stores the most recently received measurement of each physical quantity f_k. Upon receiving a request for the value of f_k at slot t (i.e., $r_k(t) = 1$), the edge node can either command sensor k to perform a new measurement and send a *status update*[2] or use the previous measurement from the local cache, to serve the request. Let $a_k(t) \in \{0, 1\}$ denote the *command* action of the edge node at slot t; $a_k(t) = 1$ if the edge node commands sensor k to send a status update and $a_k(t) = 0$ otherwise.

We assume that all the requests that arrive at the beginning of slot t are handled during the same slot t. Note that while the communications between the edge node and the users are assumed to be error-free,[3] the transmissions from the sensors to the edge node are prone to errors as detailed in Section II-C.

B. Energy Harvesting Sensors

We assume that the sensors rely on the energy harvested from the environment. Sensor k stores the harvested energy into a battery of finite size B_k (units of energy). Formally, let $b_k(t)$ denote the battery level of sensor k at the beginning of slot t. Thus, $b_k(t) \in \{0, \ldots, B_k\}$.

We consider a common assumption (see e.g., [20], [22], [23], [28], [37]) that transmitting a status update from each sensor to the edge node consumes *one* unit of energy. Once sensor k is commanded by the edge node (i.e., $a_k(t) = 1$), sensor k sends a status update if it has at least one unit of energy in its battery (i.e., $b_k(t) \geq 1$). Let random variable $d_k(t) \in \{0, 1\}$ denote the action of sensor k at slot t; $d_k(t) = 1$ if sensor k sends a status update to the edge node

[2]In general, a status update packet contains the measured value of a monitored process and a time stamp representing the time when the sample was generated.

[3]This assumption is invoked by the fact that the edge node accesses to sufficient power (e.g., a base station connected to a fixed power grid), whereas the sensors rely only on the energy harvested from the environment. However, it would be straightforward to extend our proposed approaches to the case where these links are also error-prone.

TABLE I
A COMPARATIVE SUMMARY OF CONTRIBUTIONS OF THE EXISTING WORKS IN CONTRAST TO OUR PAPER

Feature \ Ref	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]	[32]	[34]	Our
On-demand AoI	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
Cache-enabled network controller	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Partial battery knowledge	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Multiple sensors	✗	✗	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓
Multiple users	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
Energy harvesting	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✗	✓
MDP modeling	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Unreliable channel	✗	✓	✗	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓

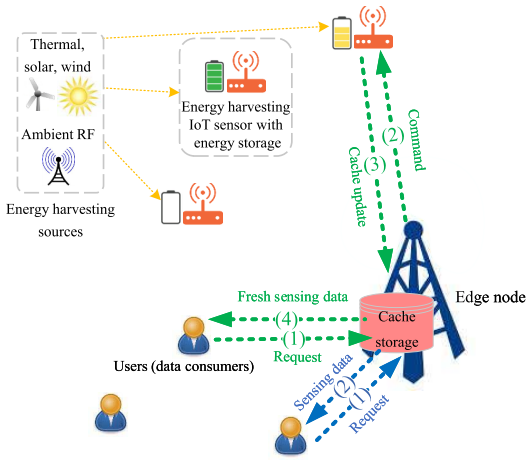


Fig. 1. An IoT sensing network consisting of multiple users (*data consumers*), one edge node (i.e., the gateway), and a set of K energy harvesting wireless IoT sensors (*data producers*). The procedure of serving a request by using fresh data is shown by green lines, whereas the blue lines show the procedure of serving a request by using the previous measurements already existing in the cache.

and $d_k(t) = 0$ otherwise. Accordingly, the relation between the action of sensor k (i.e., $d_k(t)$) and the command action of the edge node (i.e., $a_k(t)$) can be expressed as

$$d_k(t) = a_k(t)\mathbb{1}_{\{b_k(t)\geq 1\}}, \tag{1}$$

Note that quantity $d_k(t)$ in (1) characterizes also the energy consumption of sensor k at slot t.

We model the energy arrivals at the sensors as independent Bernoulli processes with intensities λ_k, $k \in \mathcal{K}$. This characterizes the discrete nature of the energy arrivals in a slotted-time system, i.e., at each time slot, a sensor either harvests one unit of energy or not (see e.g., [31]). Let $e_k(t) \in \{0, 1\}$, $t = 1, 2, \ldots$, denote the *energy arrival process* of sensor k. Thus, the probability that sensor k harvests one unit of energy during one time slot is λ_k, i.e., $\Pr\{e_k(t) = 1\} = \lambda_k$, $k \in \mathcal{K}$, $t = 1, 2, \ldots$.

Finally, using the defined quantities $b_k(t)$, $d_k(t)$, and $e_k(t)$, the evolution of the battery level of sensor k is expressed as

$$b_k(t+1) = \min\left\{b_k(t) + e_k(t) - d_k(t), B_k\right\}. \tag{2}$$

C. Communication Between the Edge Node and the Sensors

We consider an *error-free* binary/single-bit *command* link from the edge node to each sensor [21], [32], and an *error-prone* wireless communication link from each sensor to the edge node, as illustrated in Fig. 2. If a sensor sends a status update packet to the edge node, the transmission through the wireless link can be either *successful* or *failed*. Let $h_k(t) = 1$ denote the event that a status update from sensor k has been successfully received by the edge node at slot t. Otherwise, $h_k(t) = 0$ which accounts for both the cases that either 1) sensor k sends a status update but the transmission is failed, or 2) the sensor does not send a status update. Let ξ_k be the conditional probability that given that sensor k transmits a status update, it is successfully received by the edge node, i.e., $\Pr\{h_k(t) = 1 \mid d_k(t) = 1\} = \xi_k$, $k \in \mathcal{K}$, $t = 1, 2, \ldots$. Thus, ξ_k represents the *transmit success probability* of the link from sensor k to the edge node.

D. Age of Information

Age of information (AoI) is a destination-centric metric that quantifies the freshness of information of a remotely observed random process [7]–[9]. Formally, let $\Delta_k(t)$ be the AoI about the physical quantity f_k at the edge node at the beginning of slot t, i.e., the number of time slots elapsed since the generation of the most recently received status update packet from sensor k. Let $u_k(t)$ denote the most recent time slot in which the edge node received a status update packet from sensor k, i.e., $u_k(t) = \max\{t' | t' < t, h_k(t') = 1\}$; thus, the AoI about f_k can be written as the random process $\Delta_k(t) = t - u_k(t)$. We make a common assumption (see e.g., [26], [27], [30]) that $\Delta_k(t)$ is upper-bounded by a finite value $\Delta_{k,\max}$, i.e., $\Delta_k(t) \in \{1, 2, \ldots, \Delta_{k,\max}\}$. This is reasonable, because once $\Delta_k(t)$ reaches a high value $\Delta_{k,\max}$, the available measurement about physical process f_k becomes excessively stale/expired, so further counting would be irrelevant.

At each time slot, the AoI either drops to one if the edge node receives a status update from the corresponding sensor, or increases by one otherwise. Accordingly, the evolution of $\Delta_k(t)$ can be written as

$$\Delta_k(t+1) = \begin{cases} 1, & \text{if } h_k(t) = 1, \\ \min\{\Delta_k(t) + 1, \Delta_{k,\max}\}, & \text{if } h_k(t) = 0, \end{cases} \tag{3}$$

which can be expressed compactly as $\Delta_k(t+1) = \min\left\{\left(1 - h_k(t)\right)\Delta_k(t) + 1, \Delta_{k,\max}\right\}$.

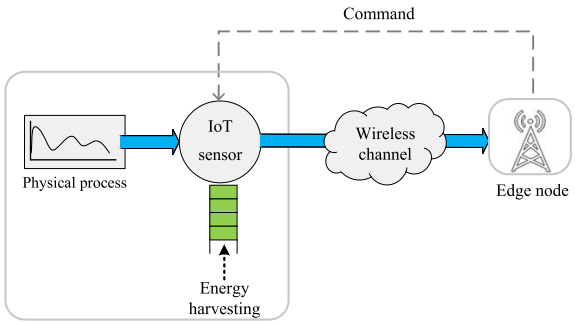


Fig. 2. The link between each sensor and the edge node consists of an error-free binary command link from the edge node to each sensor and an error-prone wireless communication link from each sensor to the edge node.

E. Cost Function and Problem Formulation

We consider a cost function that penalizes the staleness of the requested measurements received by the users. We define the per-sensor immediate cost at slot t as the *on-demand* AoI as

$$c_k(t) = r_k(t)\beta_k\Delta_k(t+1), \tag{4}$$

where $\beta_k \geq 0$ is a pre-defined weight parameter accounting for the importance of the freshness of physical quantity f_k, and $\Delta_k(t+1)$ is the AoI defined in (3). Note that when the value of f_k is not requested at slot t, i.e., $r_k(t) = 0$, the immediate cost becomes $c_k(t) = 0$, as desired. Moreover, since the requests come at the beginning of slot t and the edge node sends values to the users at the end of the same slot, $\Delta_k(t+1)$ is the effective AoI about f_k seen by the users.

We aim to find the best action of the edge node at each time slot, i.e., $a_k(t)$, $t = 1, 2, \ldots$, $k \in \mathcal{K}$, called an *optimal policy*, that minimizes the long-term average cost, defined as

$$\bar{C} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{K} c_k(t). \tag{5}$$

In order to shed light on the search for such an optimal policy, we next present several points regarding the problem structure. First, recall from Section II-A that in order to serve the requests for the value of f_k at slot t (i.e., $r_k(t) = 1$), the edge node can either command sensor k to send a status update, i.e., $a_k(t) = 1$, or use the available data in the cache, i.e., $a_k(t) = 0$. The former action (i.e., $a_k(t) = 1$), depending on the battery of sensor k and the situation of the communication link between sensor k and the edge, *may* lead to having a fresh measurement (i.e., the AoI drops to one $\Delta_k(t+1) = 1$, minimizing the immediate cost $c_k(t)$ in (4)), yet at the cost of consuming one unit of energy from the battery of sensor k. On the other hand, the latter action (i.e., $a_k(t) = 0$) provides energy saving at the cost of serving the requests by stale data. This introduces an inherent *trade-off* between (myopically) minimizing the immediate cost or saving energy for the possible future requests to minimize the cost in a long run.

It is easy to verify that if there are no requests for the value of f_k at slot t (i.e., $r_k(t) = 0$), the optimal action $a_k(t)$ that minimizes the long-term average cost (5) is $a_k(t) = 0$. In this case, the immediate cost (4) becomes $c_k(t) = 0$, and furthermore, the command action $a_k(t) = 0$ implies $d_k(t) = 0$ as per (1), leading to energy saving for sensor k. Therefore, the search for an optimal policy boils down to finding the optimal actions $a_k(t)$ for the cases with $r_k(t) = 1$.

Remark 1: For the sake of presentation, we first consider the case where the sensors have independent communication links to the edge node. Accordingly, the edge node can command any number of sensors at each slot t, and these command actions $a_k(t)$, $k \in \mathcal{K}$, are independent across k. Thus, the problem of finding the optimal actions $a_k(t)$, $k \in \mathcal{K}$, that minimize (5) is *separable* across sensors $k \in \mathcal{K}$. Then, in Section VI, we address the case where the edge node can command only a limited number of sensors, which builds on the decoupled case.

Based on Remark 1, we express the cost in (5) equivalently as $\bar{C} = \sum_{k=1}^{K} \bar{C}_k$, where $\bar{C}_k$ is the average cost associated with sensor k, i.e., the *per-sensor* long-term average cost, defined as

$$\bar{C}_k = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} c_k(t), \quad k = 1, \ldots, K. \tag{6}$$

Thus, minimizing the system-wise cost in (5) reduces to minimizing the K per-sensor long-term average costs in (6). This will be a key factor in developing our reinforcement learning (RL) algorithms in Section IV. Prior to this, in Section III, we model the considered problem as a Markov decision process (MDP) and give definitions of optimal policies, which are needed in our algorithm development.

III. Markov Decision Process and Optimal Policies

Based on Remark 1, the problem of finding an optimal policy that minimizes the long-term cost in (5) is separable across the sensors. Thus, we present the derivation of such an optimal policy for a particular sensor k but, clearly, the derivations are valid for any sensor $k \in \mathcal{K}$; the edge node runs in parallel one policy for each sensor in the network. First, we model the problem as an MDP. Then, we give a formal definition of an optimal policy, followed by introducing the key quantities needed to evaluate and search for such an optimal policy. All these serve as preliminaries for the development of our RL-based algorithms in Section IV and Section VI.

A. MDP Modeling

The MDP model associated with sensor k is defined by the tuple $\{\mathcal{S}_k, \mathcal{A}_k, \mathcal{P}_k\big(s_k(t+1)\big|s_k(t), a_k(t)\big), c_k\big(s_k(t), a_k(t)\big), \gamma\}$, where

- $\mathcal{S}_k$ is the state set. Let $s_k(t) \in \mathcal{S}_k$ denote the state at slot t, which is defined as $s_k(t) = \{b_k(t), \Delta_k(t)\}$, where 1) $b_k(t)$ is the battery level of sensor k given by (2), i.e., $b_k(t) \in \{1, 2, \ldots, B_k\}$, and 2) $\Delta_k(t)$ is the AoI about the physical quantity f_k in the local cache, i.e., $\Delta_k(t) \in \{1, 2, \ldots, \Delta_{k,\max}\}$.
- $\mathcal{A}_k = \{0, 1\}$ is the action set. The action selected by the edge node at slot t is denoted by $a_k(t) \in \mathcal{A}_k$ (see Section II-A).

- $\mathcal{P}_k\big(s_k(t+1)\big|s_k(t), a_k(t)\big)$ is the state transition probability that maps a state-action pair at slot t onto a distribution of states at slot $t+1$.
- $c_k(s_k(t), a_k(t))$ is the immediate cost function, i.e., the cost of taking action $a_k(t)$ in state $s_k(t)$, which is also denoted simply by $c_k(t)$, and is calculated using (4).
- $\gamma \in [0,1]$ is a discount factor used to weight the immediate cost relative to the future costs.

B. Optimal Policy

In an MDP environment, the immediate and long-term costs that the agent – the edge node – expects to receive depends on what actions the edge node takes at each time slot, which are selected based on a *policy*. Generally, policies can be *stochastic* or *deterministic* [38, Sect. 1.3]. A stochastic policy $\pi_k = \pi_k(a|s) : \mathcal{S}_k \times \mathcal{A}_k \rightarrow [0,1]$ is defined as a mapping from state $s \in \mathcal{S}_k$ to a *probability* of choosing each possible action $a \in \mathcal{A}_k$. A deterministic policy is a special case of the stochastic policy where in each state $s \in \mathcal{S}_k$, $\pi_k(a|s) = 1$ for some $a \in \mathcal{A}_k$. Herein, we use the same notation π_k for both stochastic and deterministic policies.

The discounted long-term accumulated cost is defined as

$$C_k(t) = \sum_{\tau=0}^{\infty} \gamma^{\tau} c_k(t+\tau), \tag{7}$$

where $c_k(\cdot)$ is the immediate cost calculated using (4). Our goal is to find an optimal policy π_k^* that minimizes the expected long-term cost in (7), defined as

$$\pi_k^* = \arg\min_{\pi_k} \mathbb{E}_{\pi_k}\left[C_k(t) \mid \pi_k\right], \tag{8}$$

where $\mathbb{E}_{\pi_k}[\cdot]$ denotes the expected value of $C_k(t)$ given that the edge node follows policy π_k.

Having defined an optimal policy, we now present essential definitions as a means to *search* for such an optimal policy.

C. State-Value and Action-Value Functions

In order to evaluate policies and search for an optimal policy π_k^*, we define the *state-value* and *action-value* functions. The state-value function specifies how beneficial it is for the edge node to be in a particular state under a policy π_k. Formally, the state-value function of state $s \in \mathcal{S}_k$ under a policy π_k can be written as

$$v_{\pi_k}(s) \doteq \mathbb{E}_{\pi_k}\left[C_k(t)|s_k(t) = s\right], \quad \forall s \in \mathcal{S}_k. \tag{9}$$

The action-value function specifies how beneficial it is for the edge node to perform a particular action in a state under a policy π_k. Formally, the action-value function can be written as

$$q_{\pi_k}(s,a) \doteq \mathbb{E}_{\pi_k}[C_k(t)|s_k(t)=s, a_k(t)=a], \quad \forall s \in \mathcal{S}_k, \ a \in \mathcal{A}_k. \tag{10}$$

Value functions define a partial ordering over policies. More precisely, a policy π_k is defined to be better than or equal to a policy π_k' (i.e., $\pi_k \geq \pi_k'$) if and only if $v_{\pi_k}(s) \leq v_{\pi_k'}(s)$ for all $s \in \mathcal{S}_k$ [38, Sect. 3.6]. Therefore, an optimal policy π_k^* (not necessarily unique), which is better than or equal to all other policies, minimizes the state-value function for all states. Optimal policies achieve the same state-value function (i.e., the *optimal state-value* function) that is defined as

$$v_k^*(s) \doteq \min_{\pi_k} v_{\pi_k}(s), \quad \forall s \in \mathcal{S}_k. \tag{11}$$

The optimal policies also share the same action-value function (i.e., the *optimal action-value* function) that is defined as

$$q_k^*(s,a) \doteq \min_{\pi_k} q_{\pi_k}(s,a), \quad \forall s \in \mathcal{S}_k, \ a \in \mathcal{A}_k. \tag{12}$$

Accordingly, an optimal deterministic policy π_k^* can be obtained by choosing the action a that minimizes $q_k^*(s,a)$ in each state s, which can expressed as

$$\pi_k^*(a|s) = \begin{cases} 1, & \text{if } a = \arg\min\limits_{a \in \mathcal{A}_k} q_k^*(s,a) \\ 0, & \text{otherwise} \end{cases}, \quad \forall s \in \mathcal{S}_k. \tag{13}$$

According to (13), the knowledge of the optimal action-value function $q_k^*(s,a)$ suffices to find an optimal policy π_k^*. Also, an optimal policy π_k^* can be found via the optimal state-value function $v_k^*(s)$, provided that the state transition probabilities are known. In this case, we first find optimal action-value function $q_k^*(s,a)$, given that $v_k^*(s)$ is available for all the states, and then find an optimal policy using (13). More precisely, under an optimal policy π_k^*, for any state $s \in \mathcal{S}_k$ and its possible successor states $s' \in \mathcal{S}_k$, the relationship between the optimal state-value and action-value functions can be derived as

$$q_k^*(s,a) = \sum_{s' \in \mathcal{S}_k} \mathcal{P}_k(s'|s,a)\left[c_k(s,a) + \gamma v_k^*(s')\right], \quad \forall s \in \mathcal{S}_k, \quad \forall a \in \mathcal{A}_k. \tag{14}$$

In summary, one can find an optimal policy if either 1) the optimal action-value function $q_k^*(s,a)$ is available, or 2) the optimal state-value function $v_k^*(s)$ and state transition probabilities $\mathcal{P}_k\big(s'|s,a\big)$ are available. We next discuss how to find $v_k^*(s)$ and $q_k^*(s,a)$.

Under π_k^*, the recursive relationship between the optimal state-value function of state s, $v_k^*(s)$, and the optimal state-value function of its possible successor state s', $v_k^*(s')$, is given by

$$\begin{aligned} v_k^*(s) &= \min_{a \in \mathcal{A}_k} q_k^*(s,a) \\ &= \min_{a \in \mathcal{A}_k} \sum_{s' \in \mathcal{S}_k} \mathcal{P}_k(s'|s,a)\left[c_k(s,a) + \gamma v_k^*(s')\right], \quad \forall s \in \mathcal{S}_k. \end{aligned} \tag{15}$$

The recursive equation in (15) is called the Bellman optimality equation for $v_k^*(s)$.

Assuming the availability of the state transition probabilities $\mathcal{P}_k(s'|s,a)$, (15) can be used to estimate the optimal state-value function recursively; this is the basis for our proposed VIA in Section IV-A. Similar to (15), the Bellman

optimality equation for $q_k^*(s,a)$ is expressed as

$$q_k^*(s,a)=\sum_{s'\in\mathcal{S}_k}\mathcal{P}_k(s'|s,a)\left[c_k(s,a)+\gamma\min_{a'\in\mathcal{A}_k}q_k^*(s',a')\right],\quad \forall s\in\mathcal{S}_k,\quad a\in\mathcal{A}_k. \quad (16)$$

The Bellman optimality equation in (16) is the basis for our proposed Q-learning algorithms devised in Section IV-B and Section IV-C.

IV. Reinforcement Learning Based Status Update Control Algorithms

In this section, we develop three RL-based status update control algorithms for the considered IoT network. The algorithms fall into two main categories: model-based RL and model-free RL. For the MDP model described in Section III-A, we first develop a model-based VIA relying on dynamic programming in Section IV-A, followed by proposing a model-free Q-learning algorithm in Section IV-B. As a practical consideration in Section IV-C, we redefine the state definition of the MDP to propose a Q-learning method for the scenario where the edge node is informed of the sensors' battery levels only via the status update packets. As a key advantage, the proposed algorithms are simple with low complexity of implementation, which is important in practice.

A. Value Iteration Algorithm (VIA)

Value Iteration is a model-based RL method that finds the optimal state-value function $v_k^*(s)$, and consequently, an optimal policy π_k^* by turning the Bellman optimality equation (15) into an iterative update procedure [38, Section 4.4].

1) Derivation of the State Transition Probabilities: In order to apply (15), the VIA requires the knowledge of the state transition probabilities of the MDP (see Section III-A). These are derived in the following. In the considered system model, for a given action $a_k(t)$, the state transition probabilities are functions of both EH rate λ_k and transmit success probability ξ_k, which were defined in Section II-B and II-C, respectively. The probability of transition from state $s_k(t)$ to state $s_k(t+1)$ under action $a_k(t)$ is given by (17) which is shown at the bottom of the next page.

In brief, the first three expressions (17a)–(17c), as shown at the bottom of the next page, correspond to cases where sensor k does not send a status update, which leads the AoI about f_k in the local cache to increase by one, whereas in (17d) sensor k sends a status update. In (17d), as shown at the bottom of the next page, four possible events can occur, depending on the success of the transmission attempt and the energy arrivals, characterized by ξ_k and λ_k, respectively. These cases are detailed in the following. noitemsep,nolistsep

- Case (17a): The edge node does not command sensor k (i.e., $a_k(t)=0$), and thus, the sensor does not send a status update.
- Case (17b): Similar to case (17a), but the battery of sensor k is full, and thus, there is no room left for possible harvested energy units.
- Case (17c): Sensor k is commanded, but since its battery is empty (i.e., $b_k(t)=0$), no update takes place.
- Case (17d): The edge node commands sensor k whose battery is non-empty (i.e., $b_k(t)\geq 1$); sensor k sends the status update, consuming one unit of energy.

Algorithm 1 Value Iteration Algorithm (VIA)

Initialize $v_k^*(s)=0$, $k\in\mathcal{K},\forall s\in\mathcal{S}_k$, and determine a small threshold $\theta>0$.
for $k=1,\ldots,K$ **do**
  repeat {*Update* $v_k^*(s)$}
    $\delta=0$ {*For stopping criterion*}
    for $s\in\mathcal{S}_k$ **do**
      $\nu=v_k^*(s)$
      $v_k^*(s)=\min_{a\in\mathcal{A}_k}\sum_{s'\in\mathcal{S}_k}\mathcal{P}_k(s'|s,a)\left[c_k(s,a)+\gamma v_k^*(s')\right]$
      $\pi_k^*(a|s)=\mathbb{1}_{\{a=\arg\min_{a\in\mathcal{A}_k}\sum_{s'\in\mathcal{S}_k}\mathcal{P}_k(s'|s,a)[c_k(s,a)+\gamma v_k^*(s')]\}}$
      $\delta=\max\{\delta,|\nu-v_k^*(s)|\}$ {*Maximum deviation between the iterations*}
    end for
  until $\delta<\theta$
end for
Output: Optimal deterministic per-sensor policies $\pi_k^*(a|s)$, $\forall k\in\mathcal{K}$.

2) Algorithm Summary: Having defined the state transition probabilities above, we now employ the Bellman optimality equation (15) and set up an iterative update procedure, the VIA, to find an optimal policy π_k^*. The proposed VIA is presented in Algorithm 1, which consists of four main stages: 1) an arbitrary initialization for the optimal state-value function, e.g., $v_k^*(s)=0$, $\forall s\in\mathcal{S}_k$, 2) in each iteration, update the estimated value for $v_k^*(s)$, $\forall s\in\mathcal{S}_k$, 3) stop when the maximum difference in $v_k^*(s)$ between two consecutive iterations is below a pre-defined threshold θ, and 4) determine an optimal deterministic policy $\pi_k^*(a|s)$ by using (14) and (13).

In the VIA, it is assumed that the state transition probabilities are known in advance. According to (17), in order to calculate the state transition probabilities $\mathcal{P}_k(s'|s,a)$, the probabilistic model of the environment, i.e., EH probability λ_k and the transmit success probability ξ_k need to be known, which are not always available in practice. The scenarios under *unknown* state transition probabilities are addressed in the next subsections.

B. Q-Learning Algorithm

Q-learning is an *online* model-free RL algorithm that estimates/learns the optimal action-value functions *by experience* and finds an optimal policy iteratively. The main difference to the VIA in Section IV-A is that Q-learning does not require the knowledge of the state transition probabilities $\mathcal{P}_k(s'|s,a)$.

In the Q-learning method, the estimated action-value function for sensor k, denoted as $Q_k(s,a)$, $s\in\mathcal{S}_k$, $a\in\mathcal{A}_k$, directly approximates the optimal action-value function $q_k^*(s,a)$ in (12) [38, Sect. 6.5]. The convergence $Q_k\rightarrow q_k^*$ requires that all state-action pairs continue to be updated. To satisfy this condition, a typical approach is to use the "exploration-exploitation" technique in the action selection. The ϵ-greedy algorithm is one such method that trade-offs

exploration and exploitation [38, Sect. 6.5]. Intuitively, exploration is finding more information about the environment, while exploitation is exploiting known information to minimize the long-term cost.

Our proposed Q-learning algorithm is presented in Algorithm 2. To allow exploration-exploitation, the edge node takes either a random or greedy action at slot t; the probability of taking a random action is denoted by $\epsilon(t)$, and thus, the probability of exploiting the greedy action $a_k(t) = \arg\min_{a\in\mathcal{A}_k} Q_k(s_k(t), a)$ is $1-\epsilon(t)$. Generally, during initial iterations, it is better to set $\epsilon(t)$ high in order to learn the underlying dynamics, i.e., to allow more exploration. On the other hand, in stationary settings and once enough observations are made, small values of $\epsilon(t)$ become preferable to increase tendency to exploitation.

As it is shown on line 18 in Algorithm 2, at each slot/iteration, the value for the Q-function of the current state is updated based on the action taken and the resulting next state, where $\alpha(t)$ represents the learning rate at slot t.

C. Q-Learning Algorithm With Partial Battery Knowledge

In Section III-A, we modeled the state of the MDP as $s_k(t) = \{b_k(t), \Delta_k(t)\}$. Consequently, both the proposed VIA in Section IV-A and the Q-learning algorithm in Section IV-B rely on the assumption that the edge node knows the *exact* battery levels of the sensors at *each* time slot. This requires continual coordination between the edge node and the sensors, which may not always be feasible. In this section, we consider a realistic environment where the edge node is informed about the battery levels of the sensors only via the *status update packets*. Consequently, the edge node has only *partial* knowledge about the battery levels at each time slot.

To account for the fact that the edge node is informed about the sensors' battery levels only via the status update packets, we next modify the state definition of the MDP. A status update packet generated at the beginning of slot t consists of the value of physical quantity f_k, the battery level of sensor k (i.e., $b_k(t)$), and the timestamp t when the sample

$$\mathcal{P}_k\big(s_k(t+1)\big|s_k(t) = \{b_k(t) < B_k, \Delta_k(t)\}, a_k(t) = 0\big)$$
$$= \begin{cases} \lambda_k, & s_k(t+1) = \left\{ \begin{array}{l} b_k(t+1) = b_k(t) + 1, \\ \Delta_k(t+1) = \min\{\Delta_k(t)+1, \Delta_{k,\max}\} \end{array} \right\}; \\ 1-\lambda_k, & s_k(t+1) = \left\{ \begin{array}{l} b_k(t+1) = b_k(t), \\ \Delta_k(t+1) = \min\{\Delta_k(t)+1, \Delta_{k,\max}\} \end{array} \right\}; \\ 0, & \text{otherwise.} \end{cases} \tag{17a}$$

$$\mathcal{P}_k\big(s_k(t+1)\big|s_k(t) = \{b_k(t) = B_k, \Delta_k(t)\}, a_k(t) = 0\big)$$
$$= \begin{cases} 1, & s_k(t+1) = \left\{ \begin{array}{l} b_k(t+1) = B_k, \\ \Delta_k(t+1) = \min\{\Delta_k(t)+1, \Delta_{k,\max}\} \end{array} \right\}; \\ 0, & \text{otherwise.} \end{cases} \tag{17b}$$

$$\mathcal{P}_k\big(s_k(t+1)\big|s_k(t) = \{b_k(t) = 0, \Delta_k(t)\}, a_k(t) = 1\big)$$
$$= \begin{cases} \lambda_k, & s_k(t+1) = \left\{ \begin{array}{l} b_k(t+1) = 1, \\ \Delta_k(t+1) = \min\{\Delta_k(t)+1, \Delta_{k,\max}\} \end{array} \right\}; \\ 1-\lambda_k, & s_k(t+1) = \left\{ \begin{array}{l} b_k(t+1) = 0, \\ \Delta_k(t+1) = \min\{\Delta_k(t)+1, \Delta_{k,\max}\} \end{array} \right\}; \\ 0, & \text{otherwise.} \end{cases} \tag{17c}$$

$$\mathcal{P}_k\big(s_k(t+1)\big|s_k(t) = \{b_k(t) > 0, \Delta_k(t)\}, a_k(t) = 1\big)$$
$$= \begin{cases} \lambda_k\xi_k, & s_k(t+1) = \left\{ \begin{array}{l} b_k(t+1) = b_k(t), \\ \Delta_k(t+1) = 1 \end{array} \right\}; \\ \lambda_k(1-\xi_k), & s_k(t+1) = \left\{ \begin{array}{l} b_k(t+1) = b_k(t), \\ \Delta_k(t+1) = \min\{\Delta_k(t)+1, \Delta_{k,\max}\} \end{array} \right\}; \\ (1-\lambda_k)\xi_k, & s_k(t+1) = \left\{ \begin{array}{l} b_k(t+1) = b_k(t) - 1, \\ \Delta_k(t+1) = 1 \end{array} \right\}; \\ (1-\lambda_k)(1-\xi_k), & s_k(t+1) = \left\{ \begin{array}{l} b_k(t+1) = b_k(t) - 1 \\ \Delta_k(t+1) = \min\{\Delta_k(t)+1, \Delta_{k,\max}\} \end{array} \right\}; \\ 0 & \text{otherwise.} \end{cases} \tag{17d}$$

Algorithm 2 Online Status Update Control Algorithm via Q-Learning

Initialize $Q_k(s,a)=0$, $\forall s \in \mathcal{S}_k, a \in \mathcal{A}_k$, $k \in \mathcal{K}$
for each slot $t = 1, 2, 3, \ldots$ **do**
  for $k = 1, \ldots, K$ **do**
    if $r_k(t) = 0$ **then**
      $a_k(t) = 0$
    else
      $a_k(t)$ is chosen according to the following probability
      $a_k(t) = \begin{cases} \arg\min_{a \in \mathcal{A}_k} Q(s_k(t), a), & \text{w.p. } 1-\epsilon(t) \\ \text{a random action } a \in \mathcal{A}_k, & \text{w.p. } \epsilon(t) \end{cases}$
      if $a_k(t) = 1$ **then**
        Command sensor k to send a status update packet
        if $b_k(t) > 0$ **then** $d_k(t) = 1$
        else $d_k(t) = 0$
      else $d_k(t) = 0$
      end if
    end if
    Update AoI according to (3) and calculate $c_k(t)$
  end for
  Wait for the next requests and compute $s_k(t+1), \forall k \in \mathcal{K}$
  for $k = 1, \ldots, K$ **do** {*Update the Q-tables*}
  $Q_k(s_k(t), a_k(t)) \leftarrow (1-\alpha(t))Q_k(s_k(t), a_k(t)) + \alpha(t)\big(c_k(t) + \gamma \min_{a \in \mathcal{A}_k} Q_k(s_k(t+1), a)\big)$
  end for
end for

was generated. Let $\tilde{b}_k(t)$ denote the *knowledge* about the battery level of sensor k at the edge node at slot t. Formally, $\tilde{b}_k(t) = b_k(u_k(t))$, where $u_k(t)$ represents the most recent time slot in which the edge node received a status update packet from sensor k, i.e., $u_k(t) = \max\{t' | t' < t, h_k(t') = 1\}$ (see Section II-D). Namely, at time slot t, $\tilde{b}_k(t)$ describes what the battery level of sensor k was at the beginning of the most recent time slot at which the edge node received a status update from sensor k. To stress, the edge node does not know the exact battery level of the sensors at each time slot, but it only has the *partial/outdated* knowledge based on each sensor's last update.

Based on the discussions above, we modify the state definition of the MDP defined in Section III-A as $s_k(t) = \{\tilde{b}_k(t), \Delta_k(t)\}$, thus, the state contains $\tilde{b}_k(t)$ instead of $b_k(t)$. However, this state definition makes it impossible to calculate the state transition probabilities and use the VIA. In particular, the underlying decision process is non-Markovian (i.e., not an MDP), caused by the uncertainty that exists in the wireless channel. For better clarification, consider state $s_k(t) = \{\tilde{b}_k(t), \Delta_k(t)\}$ and action $a_k(t) = 0$; the next state is $s_k(t+1) = \big\{\tilde{b}_k(t), \min\{\Delta_k(t)+1, \Delta_{k,\max}\}\big\}$ with probability one. However, given $s_k(t)$ and $a_k(t) = 1$, it is impossible to calculate the state transition probabilities without knowing the actions taken by the edge node during the last $\Delta_k(t) - 1$ slots (i.e., $a_k(t - \Delta_k(t)), \ldots, a_k(t-1)$), implying the non-Markovity in respect to the current state definition. This is because the energy consumed by the sensor is unknown during these $\Delta_k(t) - 1$ slots (in which, by definition, no update has been received); at each such slot, three indistinguishable cases might have happened: 1) the edge node commanded the sensor, but the transmission was failed, or 2) the edge node commanded the sensor and it could not send a status update because its battery was empty, or 3) the edge node did not command the sensor. While the first case consumes one unit of energy from the battery of the sensor, the second and third cases do not. This means that in order to model the underlying decision process as an MDP and be able to calculate the state transition probabilities, the *exact actions* taken by the edge node during the last $\Delta_k(t) - 1$ slots must be included in the state definition. More precisely, at slot t, the state would be defined as $s_k(t) = \big\{\tilde{b}_k(t), \Delta_k(t), a_k(t - \Delta_k(t)), \ldots, a_k(t-1)\big\}$. This, however, makes the state space grow exponentially in terms of $\Delta_k(t)$.

Despite the aforementioned non-Markovity property of the decision process, we apply the Q-learning presented in Algorithm 2 for the partial battery knowledge case with state $s_k(t) = \{\tilde{b}_k(t), \Delta_k(t)\}$. Recall that the Q-learning algorithm does not need any prior knowledge about the state transition probabilities. We will assess the performance of this Q-learning method via simulations in Section VII and show that it indeed is capable of learning the underlying environment to some extent, thereby significantly outperforming several baseline methods.

V. Structural Properties of an Optimal Policy

In this section, we analyze the properties of an optimal policy defined in (8). We first prove that the optimal state-value function has monotonic properties. Then, we exploit this monotonicity to prove that an optimal policy has a threshold-based structure with respect to the AoI for the case where the link from sensor k to the edge node is error-free, i.e., $\xi_k = 1$. For general cases, threshold-based structures are also numerically illustrated in Section VII-B.

Next, we present two propositions that are used to prove properties of an optimal policy expressed in Theorem 1.

Proposition 1: The optimal state-value function $v_k^*(s)$ is (i) non-decreasing with respect to the AoI, and (ii) non-increasing with respect to the battery level.

The proof is presented in Appendix A.

Proposition 2: For the case where the link from sensor k to the edge node is perfect, i.e., $\xi_k = 1$, the difference between the optimal action-value functions for the different actions, denoted by $\delta q_k^*(s) = q_k^*(s,1) - q_k^*(s,0)$, is non-increasing with respect to the AoI.

The proof is presented in Appendix B.

Theorem 1: For the case where the link from sensor k to the edge node is perfect, i.e., $\xi_k = 1$, an optimal policy has a threshold-based structure with respect to the AoI.

Proof: Proving that an optimal policy has a threshold-based structure with respect to the AoI is equivalent to showing that if the optimal action in state $s = \{b, \Delta\}$ is $a_k^*(s) = 1$, then for all the states $\underline{s} = \{b, \underline{\Delta}\}$, in which $\underline{\Delta} \geq \Delta$, the optimal action is $a_k^*(\underline{s}) = 1$ as well. According to Proposition 2, $q_k^*(\underline{s}, 1) - q_k^*(\underline{s}, 0) \leq q_k^*(s,1) - q_k^*(s,0)$. The optimal action in state s is $a_k^*(s) = 1$, thus

$q_k^*(s,1) - q_k^*(s,0) \leq 0$. Accordingly, $q_k^*(\underline{s},1) - q_k^*(\underline{s},0) \leq 0$, which shows that the optimal action for state $\underline{s}$ is $a_k^*(\underline{s}) = 1$. □

Besides the fact that analyzing the structures give insight to optimal policies, the inherent threshold-based structure of an optimal policy can be exploited to reduce the computational complexity of the VIA (see e.g., [39]).

VI. Status Update Control Under Transmission Limitation

So far, we assumed that the edge node can command multiple sensors without any constraints at each time slot, which implies the actions $a_k(t)$, $k \in \mathcal{K}$, to be independent across k. In this section, we address the case where the edge node can command only a limited number of sensors. Suppose that, due to limited radio resources (e.g., bandwidth), the edge node can command no more than $M < K$ sensors at each time slot. Thus, we have the per-slot transmission limitation

$$\sum_{k=1}^{K} a_k(t) \leq M, \quad \forall t. \tag{18}$$

The constraint (18) *couples* the actions $a_k(t)$, $k \in \mathcal{K}$, and thus, finding an optimal policy under the transmission constraint is *not separable* across the sensors.

We next model the problem of finding an optimal policy under the transmission constraint (18) as an MDP. By defining the state similarly as in the per-sensor MDP of Section III-A while incorporating the coupling constraint into the action set allows us to use the developed RL methods of Section IV. Due to the coupling constraint, the complexity of the solution grows exponentially by increasing the number of sensors K. Thus, as a practical consideration, we also propose a sub-optimal algorithm for which the complexity increases only linearly in K. The performance of the proposed sub-optimal solution is numerically demonstrated to be close to the optimal solution in Section VII-D.

A. MDP Modeling

The problem of finding an optimal policy under the transmission constraint is modeled as an MDP, defined by the tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}(\mathbf{s}(t+1)|\mathbf{s}(t), \mathbf{a}(t)), c(\mathbf{s}(t), \mathbf{a}(t))\}$, where

- The state set $\mathcal{S}$ is defined as $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_K$; the state space dimension is $|\mathcal{S}| = \prod_{k=1}^{K} (B_k + 1)\Delta_{k,\max}$. The state of the system at slot t is defined as $\mathbf{s}(t) = \big(s_1(t), \ldots, s_K(t)\big) \doteq \big(s_k(t)\big)_{k=1}^{K}$, where $s_k(t)$ is defined in Section III-A.
- The action set $\mathcal{A}$ is defined as $\mathcal{A} = \big\{(a_1, \ldots, a_K) \mid a_k \in \mathcal{A}_k = \{0,1\},\ \sum_{k=1}^{K} a_k \leq M\big\}$; the action space dimension is $|\mathcal{A}| = \sum_{m=0}^{M} \binom{K}{m}$. The action selected by the edge node at slot t is denoted by $\mathbf{a}(t) = \big(a_k(t)\big)_{k=1}^{K}$, where $a_k(t)$ is defined in Section III-A.
- The state transition probability $\mathcal{P}(\mathbf{s}(t+1)|\mathbf{s}(t), \mathbf{a}(t))$ is calculated as

$$\mathcal{P}\big(\mathbf{s}(t+1)\big|\mathbf{s}(t), \mathbf{a}(t)\big) = \prod_{k=1}^{K} \mathcal{P}_k\big(s_k(t+1)\big|s_k(t), a_k(t)\big), \tag{19}$$

  where $\mathcal{P}_k\big(s_k(t+1)\big|s_k(t), a_k(t)\big)$ is calculated according to (17a)–(17d).
- The immediate cost function $c(\mathbf{s}(t), \mathbf{a}(t))$, denoted simply by $c(t)$, is calculated as $c(t) = \sum_{k=1}^{K} c_k(t)$, where $c_k(.)$ is defined in Section III-A.

B. Optimal and Sub-Optimal Algorithms

1) Optimal Policy: An optimal policy under the transmission constraint can be found by following the steps in Section IV and using the developed learning methods, i.e., VIA or Q-learning. Because the state and action spaces grow exponentially with respect to the number of sensors, finding an optimal policy is tractable only for a small number of sensors. More precisely, finding an optimal policy is PSPACE-hard which is similar to NP-hard except that the space (i.e., the size of computer memory) is the main limiting factor [40] [41, Chap. 6]. The structural properties of the optimal policy – obtained by VIA – can be obtained by following the same steps as in Section V, but due to the space limitation, we omit it.

2) Sub-Optimal Policy: In order to reduce the exponential complexity due to the coupling constraint (18) and deal with practical massive IoT scenarios, we propose the following sub-optimal policy. First, we *ignore* the constraint (18), and find the optimal per-sensor policies $\pi_k^\star$, $k \in \mathcal{K}$, as discussed in Section IV, either by using VIA or Q-learning. Then, we *truncate* the scheduling policy to satisfy the constraint (18) as follows. At slot t, let $\mathcal{X}(t) = \{k \mid a_k(t) = 1, k \in \mathcal{K}\} \subseteq \mathcal{K}$ denote the set of sensors that are commanded under the optimal per-sensor policies $\pi_k^\star$, $k \in \mathcal{K}$. The truncation step separates into two cases: 1) if $|\mathcal{X}(t)| \leq M$, the edge node simply commands all of the sensors in $\mathcal{X}(t)$, and 2) otherwise, the edge node commands only the M sensors from $\mathcal{X}(t)$ that have *the largest AoI*. In this regard, the truncation policy conforms to a myopic strategy in that it prioritizes updating the sensors with the highest AoI to minimize the immediate cost.

Remark 2: For the case with no energy limitations at the source nodes, a Whittle index policy can be obtained which is asymptotically optimal and has low complexity. For instance, in [39], scheduling multiple sensors with a transmission constraint was modeled as a restless multi-armed bandit (RMAB) and a Whittle index policy was obtained. In RMAB, at each time slot, a specific subset of "arms" is selected by the decision maker [41, Chap. 6]. In order to cast our problem as an RMAB and be able to find a Whittle index policy, we first need to ensure that, for an optimal policy, *exactly* M sensors are commanded by the edge node at *each* time slot. However, it is clear that in our system model, commanding exactly M sensors at each time slot is highly sub-optimal. This is because of the energy harvesting nature of the sensors. Namely, when the battery level (or the AoI) is low, it is optimal *not* to command the sensor. Inspired by the procedure of finding a Whittle index policy [41, Chap. 6], we could start by relaxing the per-slot transmission constraint to the long-term average constraint, and decouple the problem along the sensors by using the Lagrange function. Then, by applying the constrained MDP (CMDP) concepts, we can find an optimal

TABLE II
DEFAULT SIMULATION PARAMETERS

Parameter	Value
Number of sensors (K)	3
Capacity of the batteries (B_k)	15
The weight parameters (β_k)	1.0
Discount factor (γ)	0.99
Maximum deviation error in VIA (θ)	0.001
AoI upper-bound ($\Delta_{k,\max}$)	127

policy for the relaxed decoupled problem. Here, there are two main challenges: 1) properly modifying the optimal relaxed policy to satisfy the per-slot constraint, and 2) mathematical analysis to show the above policy is asymptotically optimal. Studying these aspects will be striven for in our future work.

VII. SIMULATION RESULTS

In this section, we numerically analyze the structural properties of an optimal policy obtained by the VIA. Moreover, simulation results are presented to demonstrate the performance of the proposed VIA summarized in Algorithm 1, the proposed Q-learning algorithms – Q-learning with exact and partial battery knowledge – obtained by Algorithm 2, and the proposed algorithms under the transmission limitations – optimal and sub-optimal – developed in Section VI.

A. Simulation Setup

The simulation setup is as the following, unless otherwise stated. We consider $K = 3$ EH sensors, i.e., $\mathcal{K} = \{1, 2, 3\}$. Each sensor $k \in \mathcal{K}$ has a battery with capacity $B_k = 15$ units of energy. At each time slot, the probability that the value of f_k is requested (i.e., $r_k(t) = 1$) is denoted by p_k, i.e., $\Pr\{r_k(t) = 1\} = p_k$. We set $p_k = 0.15$, $k \in \mathcal{K}$. For the VIA, we set the threshold parameter as $\theta = 0.001$. For the Q-learning method, we set $\epsilon(t) = 0.02 + 0.98\, e^{-\epsilon_{\rm d} t}$ with decay parameter $\epsilon_{\rm d} = 10^{-7}$. The learning rate is set to $\alpha(t) = 0.5$ during the first $1/\epsilon_{\rm d} = 10^7$ slots and after that $\alpha(t) = 0.01$. Table II summarizes the default simulation parameters.

B. Structure of an Optimal Deterministic Policy

We analyze the structural properties of an optimal deterministic policy obtained by the VIA for a particular sensor, e.g., sensor 1, and investigate the effect of the EH probability λ_1 and transmit success probability ξ_1.

Fig. 3 illustrates the structure of the obtained optimal deterministic policy for different values of the EH probability λ_1 with the transmit success probability $\xi_1 = 0.9$. Each point represents a potential state of the system as a pair of values of the battery level and AoI, (b, Δ). In particular, a red circle indicates that the optimal action in a given state is that the edge node does not command the sensor (i.e., $a = 0$), and a blue square indicates that the optimal action is that the edge node commands the sensor (i.e., $a = 1$). The set of blue points is referred to as the *command region* hereinafter.

From Fig. 3(a)–(d), we observe that the optimal deterministic policy has a *threshold-based* structure with respect to the battery level and the AoI, which can be expressed as follows:

1) If the optimal action in state $s = \{b, \Delta\}$ is $a = 1$, then for all the states $s' = \{b', \Delta\}$, in which $b' \geq b$, the optimal action is $a = 1$ as well.
2) If the optimal action in state $s = \{b, \Delta\}$ is $a = 1$, then for all the states $s' = \{b, \Delta'\}$, in which $\Delta' \geq \Delta$, the optimal action is $a = 1$ as well.[4]

To exemplify this threshold-based structure in Fig. 3(a), consider point $(5, 17)$. Since the optimal action at the point $(5, 17)$ is $a = 1$, we observe that the optimal action at all the points $(5, \Delta)$ where $\Delta \geq 17$, and all the points $(b, 17)$ where $b \geq 5$, is also $a = 1$.

By comparing Figs. 3(a)–(d) with each other, we observe that the command region (i.e., the set of blue square points) enlarges by increasing the EH probability λ_1. This is due to the fact that since the sensor harvests energy more often, the edge node commands the sensor to send fresh measurements more often. Note that Fig. 3(d) is associated with an extreme case in which the edge node always harvests energy at each time slot; in this case, there is always at least one unit of energy available in the battery of the sensor, and thus, for all the states with $b \geq 1$, the optimal action is $a = 1$.

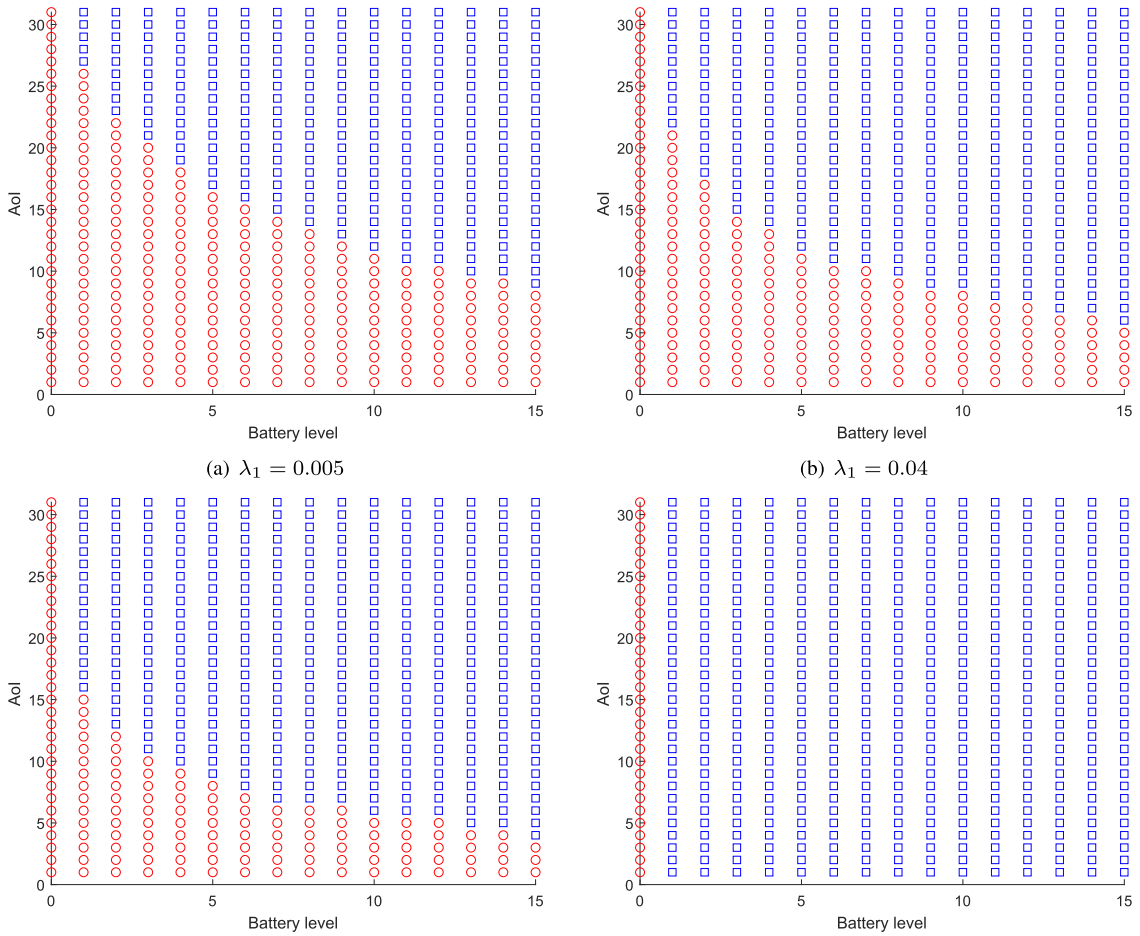
Fig. 4 illustrates the threshold-based structure of the obtained optimal deterministic policy for different values of the transmit success probability ξ_1 with the EH probability $\lambda_1 = 0.04$. Figs. 4(a)–(d) illustrate that the command region expands by increasing the transmit success probability ξ_1. This is due to the fact that by increasing ξ_1, the communication link from the sensor to the edge node becomes more reliable, and thus, the edge node commands the sensor more often as it has more confidence about receiving the transmitted status update packet. Fig. 4(a) depicts an extreme case with $\xi_1 = 0$, in which the link from the sensor to the edge node is always in the failed state and the edge node never receives any commanded status update; to conserve the sensor's battery, the optimal action is clearly $a = 0$.

C. Performance and Learning Behaviour of the Proposed Algorithms

We investigate the performance and learning behaviour of the proposed Q-learning algorithms with exact and partial battery knowledge. To this end, we analyze the performance of the proposed algorithms in terms of the long-term average costs defined in (5) and (6). As a remark, the VIA serves as a lower bound to the proposed Q-learning algorithms since it knows the exact statistical model of the environment, and consequently, the state transition probabilities of the underlying MDP. Similarly, the Q-learning method with the exact battery knowledge (referred to as *Q-learning-exact* hereinafter) is a lower bound to the Q-learning algorithm having only the partial battery knowledge (referred to as *Q-learning-partial* hereinafter).

For comparison, we consider two baseline policies: *greedy (myopic)*, *greedy-threshold*, and *random* policy. In the greedy

[4]In Section V, we analytically proved this statement for the special case $\xi_k = 1$. In this section, the numerical results show that an optimal policy has a threshold-based structure with respect to the AoI for all the values of ξ_k as well.



(a) $\lambda_1 = 0.005$ (b) $\lambda_1 = 0.04$

(c) $\lambda_1 = 0\ 08$ (d) $\lambda_1 = 1\ 0$

Fig. 3. Structure of an optimal deterministic policy π_1^* obtained by the VIA for each state $s = \{b, \Delta\}$ with the transmit success probability $\xi_1 = 0.9$ for different values of the EH probability λ_1. Red circle: no command $a = 0$; blue square: command $a = 1$.

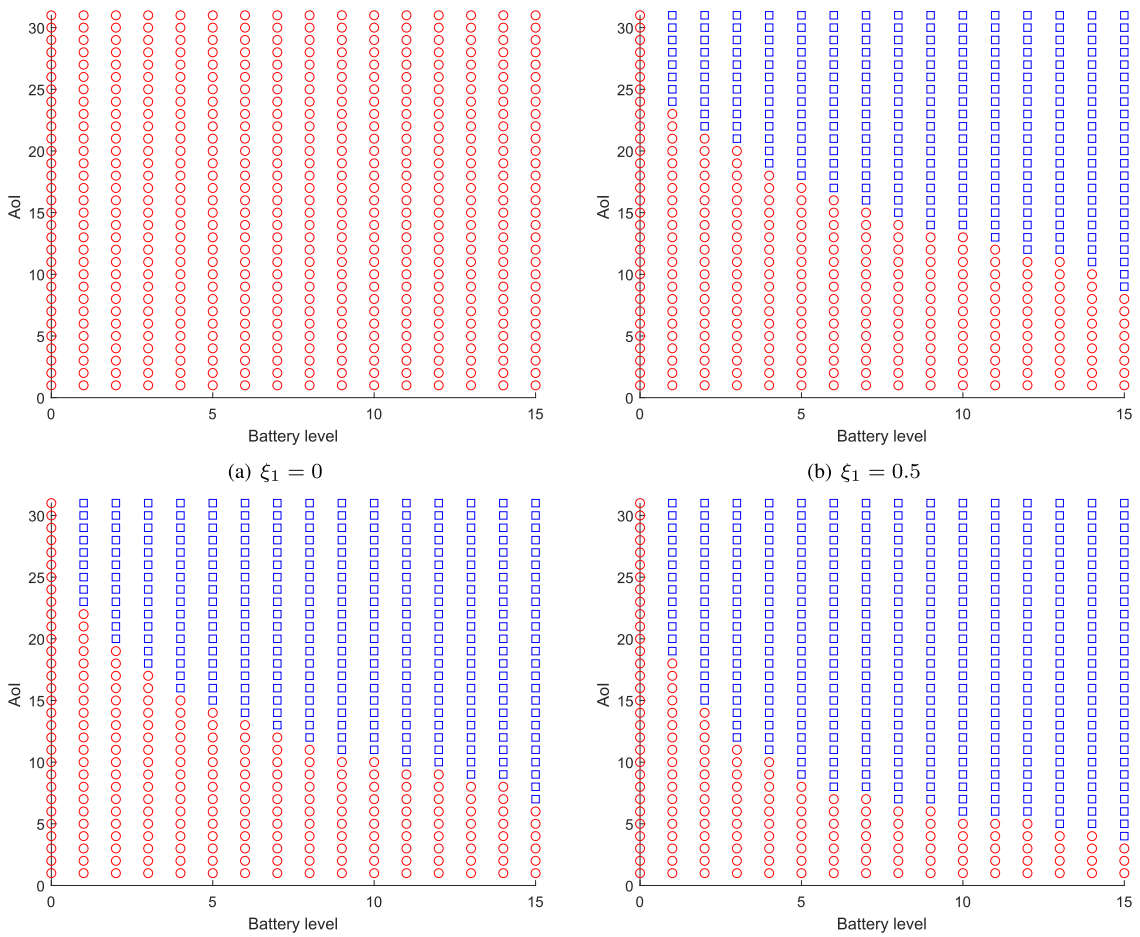
policy, whenever the value of physical quantity f_k is requested (i.e., $r_k(t) = 1$), the edge node commands sensor k (i.e., $a_k(t) = 1$), regardless of the battery stage and AoI; sensor k sends a status update if the battery is non-empty, i.e., $b_k(t) \geq 1$. In the greedy-threshold policy, whenever the value of physical quantity f_k is requested (i.e., $r_k(t) = 1$), the edge node commands sensor k if the battery level of sensor k is above a threshold $\mathrm{b_{Th}}$ (i.e., $b_k(t) \geq \mathrm{b_{Th}}$). Note that the greedy-threshold policy with $\mathrm{b_{Th}} = 1$ is equivalent to the greedy (myopic) policy. In the random policy, whenever the value of physical quantity f_k is requested (i.e., $r_k(t) = 1$), the edge node selects a random action $a_k(t) \in \{0, 1\}$ according to the discrete uniform distribution.

Fig. 5 depicts the performance of each algorithm for the EH probabilities $\lambda_1 = 0.04$, $\lambda_2 = 0.05$, and $\lambda_3 = 0.06$, and the transmit success probabilities $\xi_k = 0.15$, $\forall k \in \mathcal{K}$. Figs. 5(a)–(c) are associated with the per-sensor long-term average cost ($\bar{C}_k$) for sensor 1, 2, and 3, respectively. Fig. 5(d) illustrates the long-term average cost over all the sensors ($\bar{C}$).

As it is shown in Fig. 5(d), Q-learning-exact performs close to the VIA and the proposed RL algorithms outperform the baseline methods in terms of the long-term average cost. The figures show that among the greedy-threshold baseline policies, the greedy (myopic) policy ($\mathrm{b_{Th}} = 1$) results in the best performance. Q-learning-exact, and also the VIA, reduce the average cost approximately by a factor of 2 compared to the greedy algorithm. Furthermore, the average cost decreases roughly 30 % for Q-learning-partial compared to the (myopic) greedy algorithm.

Interestingly, the gap between Q-learning-partial and Q-learning-exact is small, when the EH probability is high enough. As it is shown in Figs. 5(a)–(c), the largest gap occurs for the sensor with the lowest EH probability, i.e., sensor 1; on the contrary, the smallest gap is obtained for sensor 3 having the highest EH probability. This is due to the fact that when the energy becomes scarce, the edge node receives status updates more rarely; consequently, the information about the battery levels at the edge node becomes more outdated, i.e., more uncertain, inhibiting the capability of Q-learning-partial to take near-optimal actions as taken by Q-learning-exact. Overall, Fig. 5 demonstrates that the proposed algorithm for a realistic scenario has high performance even if the edge node performs actions based on the outdated battery information.

In Fig. 5(a), the greedy policy performs as poorly as the random policy, because the EH probability is low, and thus, it is highly sub-optimal to command the sensor at all states. As it can be seen in Figs. 5(a)–(c), the lowest long-term average



(a) $\xi_1 = 0$ (b) $\xi_1 = 0.5$

(c) $\xi_1 = 0.7$ (d) $\xi_1 = 1.0$

Fig. 4. Structure of an optimal deterministic policy π_1^* obtained by the VIA for each state $s = \{b, \Delta\}$ with the EH probability $\lambda_1 = 0.04$ for different values of the transmit success probability ξ_1. Red circle: no command $a = 0$; blue square: command $a = 1$.

cost is associated with the sensor that has the highest EH probability, i.e., sensor 3. This is because sensor 3 harvests energy more often, and thus, it can send status updates more frequently upon receiving a command from the edge node. Recall that the command region enlarges by increasing the EH probability, i.e., the edge node commands the sensor more frequently.

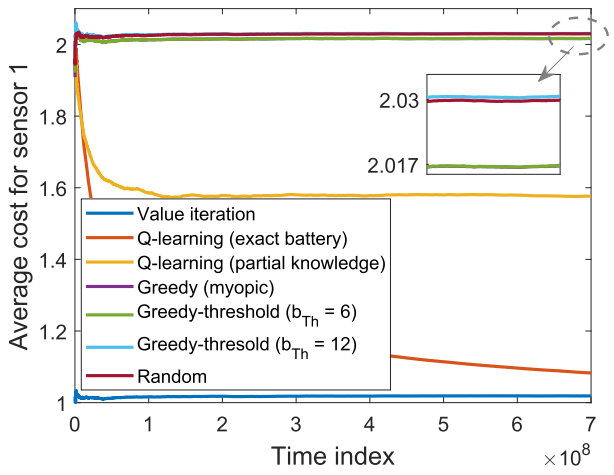
By comparing Figs. 5(a)–(c) with each other, we observe that by increasing the EH probability λ_k the long-term average cost for the VIA, and also for the Q-learning, moves toward the long-term average cost for the greedy policy. This is because by increasing the EH probability, the command region enlarges, and thus, an optimal policy tends to the greedy policy.

D. Performance Under the Transmission Constraint

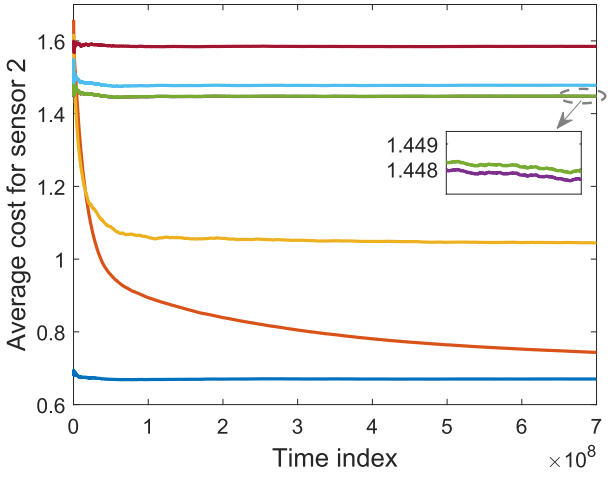
We investigate the performance of the proposed optimal and sub-optimal solutions presented in Section VI. The results are obtained by averaging each algorithm over 200 episodes whereas each episode takes 10^6 slots. We compare the proposed policy with the greedy and random policies. In the greedy policy, due to transmission constraint, the edge node commands no more than M sensors with the largest AoI from the set $\mathcal{W}(t) = \{r_k(t) = 1, k \in \mathcal{K}\}$ (i.e., the set of sensors whose measurements are requested by user(s)). In Fig. 6(a), the performance of the optimal and sub-optimal policies are compared for different values of the transmission constraint parameter M in a simple scenario with $K = 4$, $B_k = 4$, $\Delta_{k,\max} = 8$, and $p_k = 1$. As shown, the gap between the proposed optimal and sub-optimal solutions is small, even though the complexity of the sub-optimal is significantly lower than that of the optimal solution, as discussed in Section VI. In Fig. 6(b), a more realistic scenario is considered in which $K = 25$, $B_k = 7$, $\Delta_{k,\max} = 64$, $p_k = 1$. Note that running our algorithm to find an optimal policy in this scenario is not tractable because the state space dimension is $|\mathcal{S}| \approx 5 \times 10^{67}$. For the benchmarking, we also plot the optimal policy for the case without any transmission constraint to serve as a lower bound. As shown, the performance of sub-optimal policy is close to the lower bound for $M \geq 2$, which shows the effectiveness of the proposed sub-optimal solution. Furthermore, the sub-optimal policy yields roughly $50\,\%$ lower average cost than the baseline methods for (almost) all values of M.

VIII. Conclusion and Future Work

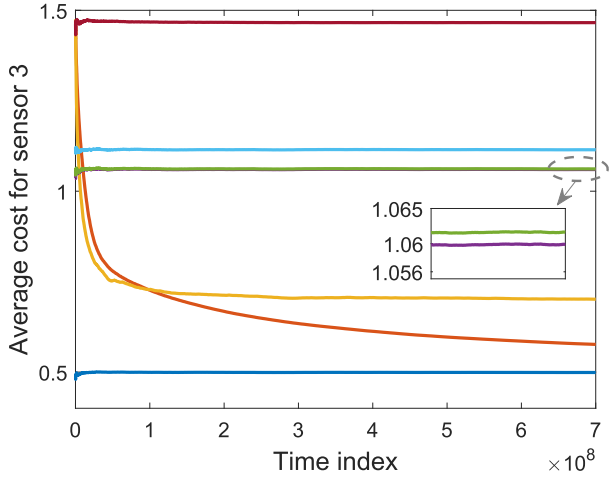
We investigated a status update control problem in an IoT sensing network consisting of multiple users, multiple EH



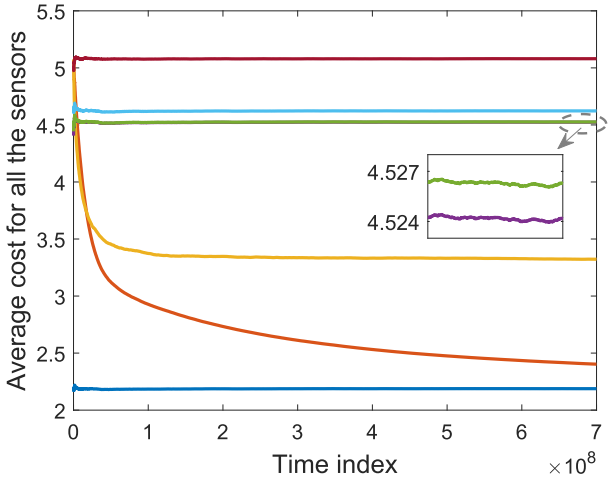
(a) Average cost for sensor 1, $\bar{C}_1$, with $\lambda_1 = 0.04$ and $\xi_1 = 0.15$



(b) Average cost for sensor 2, $\bar{C}_2$, with $\lambda_2 = 0.05$ and $\xi_2 = 0.15$

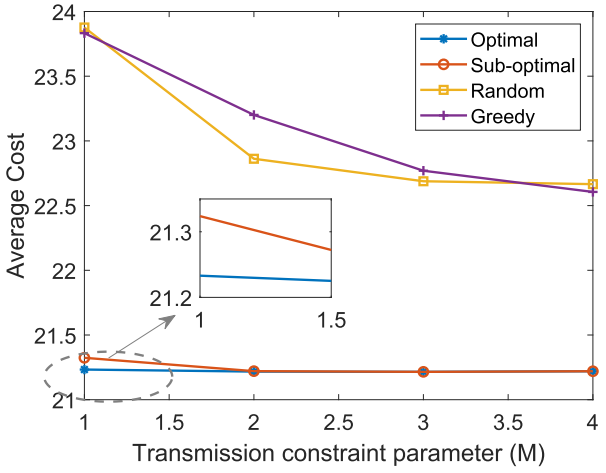


(c) Average cost for sensor 3, $\bar{C}_3$, with $\lambda_3 = 0.06$ and $\xi_3 = 0.15$

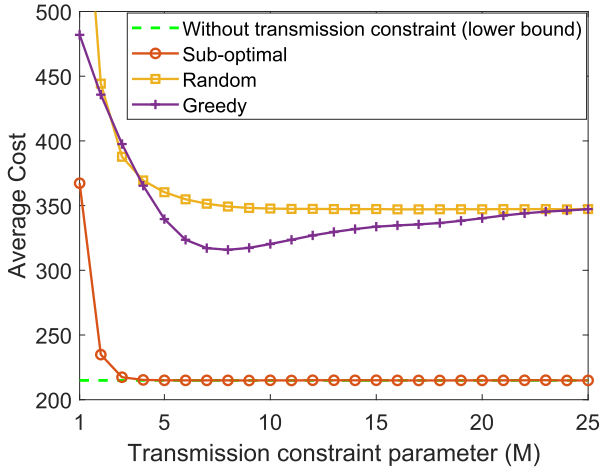


(d) Average cost for all the sensors $\bar{C}$

Fig. 5. Learning behaviour of the proposed VIA and Q-learning algorithms in comparison to baseline policies.



(a) Number of sensors $K = 4$



(b) Number of sensors $K = 25$

Fig. 6. Performance of the proposed optimal and sub-optimal policies under the transmission limitation in comparison to the baseline policies.

sensors, and a wireless edge node. We modeled the problem as an MDP and proposed two classes of RL based algorithms: a model-based VIA relying on dynamic programming, and a model-free Q-learning method. Furthermore, we developed a Q-learning method for the realistic case in which the edge node does not know the exact battery levels. The proposed Q-learning schemes do not need any information about the EH model. We also proposed an optimal and a low-complexity

sub-optimal algorithm for a massive IoT scenario where the edge node can command only a limited number of sensors. Simulation results showed that an optimal policy has a threshold-based structure and the proposed RL algorithms significantly reduce the long-term average cost compared to several baseline methods.

Interesting future direction of this work would be to investigate the case where the edge node cannot serve the requests from all the users at one time slot, and study the impact of user scheduling on the age-optimal policies for the large-scale EH IoT networks. Another future direction could be to search for optimal and/or low-complexity algorithms under both the partial battery knowledge at the edge node and the transmission limitation.

APPENDIX A
PROOF OF PROPOSITION 1

Proof: As discussed in Section IV-A, the optimal state-value function $v_k^*(s)$ can be computed iteratively by the VIA. In the VIA, the optimal state-value function of state s at iteration $n = 1, 2, \ldots$, denoted by $v_k^*(s)^{(n)}$, is updated as (see (15))

$$\begin{aligned} v_k^*(s)^{(n)} &= \min_{a\in\mathcal{A}_k} \sum_{s'\in\mathcal{S}_k} \mathcal{P}_k(s'|s,a)\left[c_k(s,a) + \gamma v_k^*(s')^{(n-1)}\right] \\ &= \min_{a\in\mathcal{A}_k} q_k^*(s,a)^{(n-1)}, \quad \forall s \in \mathcal{S}_k. \end{aligned} \tag{20}$$

Thus, an optimal policy at nth iteration is given by $\pi_k^*(a|s)^{(n)} = \mathbb{1}_{\{a=\arg\min_{a\in\mathcal{A}_k} q_k^*(s,a)^{(n)}\}}$. Accordingly, an optimal action in state s at nth iteration, denoted by $a_k^*(s)^{(n)}$, reads as

$$a_k^*(s)^{(n)} = \arg\min_{a\in\mathcal{A}_k} q_k^*(s,a)^{(n)}. \tag{21}$$

For any arbitrary initialization $v_k^*(s)^{(0)}$, the sequence $\{v_k^*(s)^{(n)}\}$ can be shown to converge to the optimal state-value function $v_k^*(s)$ [38, Sect. 4.4], i.e.,

$$\lim_{n\to\infty} v_k^*(s)^{(n)} = v_k^*(s). \tag{22}$$

(i) In order to prove that $v_k^*(s)$ is non-decreasing with respect to the AoI, we define two states $s = \{b, \Delta\}$ and $\underline{s} = \{b, \underline{\Delta}\}$, where $\underline{\Delta} \geq \Delta$, and show that $v_k^*(\underline{s}) \geq v_k^*(s)$. According to (22), it suffices to prove that $v_k^*(\underline{s})^{(n)} \geq v_k^*(s)^{(n)}$, $\forall n$. We prove this by mathematical induction. The initial values can be chosen arbitrarily, e.g., $v_k^*(s)^{(0)} = 0$ and $v_k^*(\underline{s})^{(0)} = 0$, thus, the relation $v_k^*(\underline{s})^{(n)} \geq v_k^*(s)^{(n)}$ holds for $n = 0$. Assume that $v_k^*(\underline{s})^{(n)} \geq v_k^*(s)^{(n)}$ for some n. We need to prove that $v_k^*(\underline{s})^{(n+1)} \geq v_k^*(s)^{(n+1)}$ as well. From (20) and (21), we have

$$\begin{aligned} &v_k^*(s)^{(n+1)} - v_k^*(\underline{s})^{(n+1)} \\ &\quad= \min_{a\in\mathcal{A}_k} q_k^*(s,a)^{(n)} - \min_{a\in\mathcal{A}_k} q_k^*(\underline{s},a)^{(n)} \\ &\quad= q_k^*\big(s, a_k^*(s)^{(n)}\big)^{(n)} - q_k^*\big(\underline{s}, a_k^*(\underline{s})^{(n)}\big)^{(n)} \\ &\quad\overset{(a)}{\leq} q_k^*\big(s, a_k^*(\underline{s})^{(n)}\big)^{(n)} - q_k^*\big(\underline{s}, a_k^*(\underline{s})^{(n)}\big)^{(n)}, \end{aligned} \tag{23}$$

where (a) follows from the fact that taking action $a_k^*(\underline{s})^{(n)}$ in state s is not necessarily optimal. We show that $q_k^*\big(s, a_k^*(\underline{s})^{(n)}\big)^{(n)} - q_k^*\big(\underline{s}, a_k^*(\underline{s})^{(n)}\big)^{(n)} \leq 0$ for all possible actions $a_k^*(\underline{s})^{(n)} \in \{0, 1\}$. We present the proof for the case corresponding to (17d) where $b \geq 1$ and $a_k^*(\underline{s})^{(n)} = 1$; for the other three cases (17a)–(17c), the proof follows similarly. We have the relations in (24), shown at the bottom of the page, where in step (a) we use the result of (17d), step (b) follows from the assumption $\Delta \leq \underline{\Delta}$, and steps (c) and (d) follow from the induction assumption.

$$\begin{aligned} &q_k^*(s,1)^{(n)} - q_k^*(\underline{s},1)^{(n)} \\ &= \sum_{s'\in\mathcal{S}_k} \mathcal{P}_k(s'|s,1)\left[c_k(s,1) + \gamma v_k^*(s')^{(n)}\right] - \sum_{\underline{s}'\in\mathcal{S}_k} \mathcal{P}_k(\underline{s}'|\underline{s},1)\left[c_k(\underline{s},1) + \gamma v_k^*(\underline{s}')^{(n)}\right] \\ &\overset{(a)}{=} \lambda_k\xi_k\left(1 + \gamma v_k^*(b,1)^{(n)}\right) + (1-\lambda_k)\xi_k\left(1 + \gamma v_k^*(b-1,1)^{(n)}\right) \\ &+ \lambda_k(1-\xi_k)\left(\min\{\Delta+1, \Delta_{k,\max}\} + \gamma v_k^*(b, \min\{\Delta+1, \Delta_{k,\max}\})^{(n)}\right) \\ &+ (1-\lambda_k)(1-\xi_k)\left(\min\{\Delta+1, \Delta_{k,\max}\} + \gamma v_k^*(b-1, \min\{\Delta+1, \Delta_{k,\max}\})^{(n)}\right) \\ &- \lambda_k\xi_k\left(1 + \gamma v_k^*(b,1)^{(n)}\right) - (1-\lambda_k)\xi_k\left(1 + \gamma v_k^*(b-1,1)^{(n)}\right) \\ &- \lambda_k(1-\xi_k)\left(\min\{\underline{\Delta}+1, \Delta_{k,\max}\} + \gamma v_k^*(b, \min\{\underline{\Delta}+1, \Delta_{k,\max}\})^{(n)}\right) \\ &- (1-\lambda_k)(1-\xi_k)\left(\min\{\underline{\Delta}+1, \Delta_{k,\max}\} + \gamma v_k^*(b-1, \min\{\underline{\Delta}+1, \Delta_{k,\max}\})^{(n)}\right) \\ &= (1-\xi_k)\underbrace{\left(\min\{\Delta+1, \Delta_{k,\max}\} - \min\{\underline{\Delta}+1, \Delta_{k,\max}\}\right)}_{(b)\leq 0} \\ &+ \gamma\lambda_k(1-\xi_k)\underbrace{\left(v_k^*(b, \min\{\Delta+1, \Delta_{k,\max}\})^{(n)} - v_k^*(b, \min\{\underline{\Delta}+1, \Delta_{k,\max}\})^{(n)}\right)}_{(c)\leq 0} \\ &+ \gamma(1-\lambda_k)(1-\xi_k)\underbrace{\left(v_k^*(b-1, \min\{\Delta+1, \Delta_{k,\max}\})^{(n)} - v_k^*(b-1, \min\{\underline{\Delta}+1, \Delta_{k,\max}\})^{(n)}\right)}_{(d)\leq 0} \leq 0. \end{aligned} \tag{24}$$

$$
\begin{aligned}
&q_k^*(s,1)-q_k^*(\underline{s},1)-q_k^*(s,0)+q_k^*(\underline{s},0)\\
&=\sum_{s'\in\mathcal{S}_k}\mathcal{P}_k(s'|s,1)\Big[c_k(s,1)+\gamma v_k^*(s')\Big]-\sum_{\underline{s}'\in\mathcal{S}_k}\mathcal{P}_k(\underline{s}'|\underline{s},1)\Big[c_k(\underline{s},1)+\gamma v_k^*(\underline{s}')\Big]\\
&-\sum_{s'\in\mathcal{S}_k}\mathcal{P}_k(s'|s,0)\Big[c_k(s,0)+\gamma v_k^*(s')\Big]+\sum_{\underline{s}'\in\mathcal{S}_k}\mathcal{P}_k(\underline{s}'|\underline{s},0)\Big[c_k(\underline{s},0)+\gamma v_k^*(\underline{s}')\Big]\\
&=\lambda_k\Big(1+\gamma v_k^*(b,1)\Big)+(1-\lambda_k)\Big(1+\gamma v_k^*(b-1,1)\Big)\\
&-\lambda_k\Big(1+\gamma v_k^*(b,1)\Big)-(1-\lambda_k)\Big(1+\gamma v_k^*(b-1,1)\Big)\\
&-\lambda_k\Big(\min\{\Delta+1,\Delta_{k,\max}\}+\gamma v_k^*(b+1,\min\{\Delta+1,\Delta_{k,\max}\})\Big)\\
&-(1-\lambda_k)\Big(\min\{\Delta+1,\Delta_{k,\max}\}+\gamma v_k^*(b,\min\{\Delta+1,\Delta_{k,\max}\})\Big)\\
&+\lambda_k\Big(\min\{\underline{\Delta}+1,\Delta_{k,\max}\}+\gamma v_k^*(b+1,\min\{\underline{\Delta}+1,\Delta_{k,\max}\})\Big)\\
&+(1-\lambda_k)\Big(\min\{\underline{\Delta}+1,\Delta_{k,\max}\}+\gamma v_k^*(b,\min\{\underline{\Delta}+1,\Delta_{k,\max}\})\Big)\\
&=\underbrace{\Big(\min\{\underline{\Delta}+1,\Delta_{k,\max}\}-\min\{\Delta+1,\Delta_{k,\max}\}\Big)}_{(a)\geq 0}\\
&+\gamma\lambda_k\underbrace{\Big(v_k^*(b+1,\min\{\underline{\Delta}+1,\Delta_{k,\max}\})-v_k^*(b+1,\min\{\Delta+1,\Delta_{k,\max}\}\Big)}_{(b)\geq 0}\\
&+\gamma(1-\lambda_k)\underbrace{\Big(v_k^*(b,\min\{\underline{\Delta}+1,\Delta_{k,\max}\})-v_k^*(b,\min\{\Delta+1,\Delta_{k,\max}\})\Big)}_{(c)\geq 0}\geq 0.
\end{aligned}
\tag{25}
$$

(ii) In order to prove that $v_k^*(s)$ is non-increasing with respect to the battery level, we define two states $s=\{b,\Delta\}$ and $\underline{s}=\{\underline{b},\Delta\}$, where $\underline{b}\geq b$. By using induction and following the similar steps as we have done in (i), one can easily show that $v_k^*(s)\geq v_k^*(\underline{s})$. □

APPENDIX B
PROOF OF PROPOSITION 2

Proof: We define states $s=\{b,\Delta\}$ and $\underline{s}=\{b,\underline{\Delta}\}$, where $\underline{\Delta}\geq\Delta$. We show that $\delta q_k^*(s)\geq\delta q_k^*(\underline{s})$, which can be rewritten as $q_k^*(s,1)-q_k^*(\underline{s},1)-q_k^*(s,0)+q_k^*(\underline{s},0)\geq 0$. We present the proof for the case where $1\leq b<B_k$; for the other two cases, i.e., $b=0$ and $b=B_k$, the proof follows similarly. We have the relations in (25), shown at the top of the page, where step (a) follows from the assumption $\Delta\leq\underline{\Delta}$, and steps (b) and (c) follow from Proposition 1. □

REFERENCES

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

[2] D. Ciuonzo, P. S. Rossi, and P. Willett, "Generalized Rao test for decentralized detection of an uncooperative target," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 678–682, May 2017.

[3] D. Ciuonzo, G. Gelli, A. Pescape, and F. Verde, "Decision fusion rules in ambient backscatter wireless sensor networks," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Istanbul, Turkey, Sep. 2019, pp. 1–6.

[4] B. Ji, B. Xing, K. Song, C. Li, H. Wen, and L. Yang, "The efficient backFi transmission design in ambient backscatter communication systems for IoT," *IEEE Access*, vol. 7, pp. 31397–31408, 2019.

[5] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, 3rd Quart., 2011.

[6] S. Kim *et al.*, "Ambient RF energy-harvesting technologies for self-sustainable standalone wireless sensor platforms," *Proc. IEEE*, vol. 102, no. 11, pp. 1649–1666, Nov. 2014.

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

[8] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1807–1827, Mar. 2019.

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

[10] Y. Sun, I. Kadota, R. Talak, and E. Modiano, "Age of information: A new metric for information freshness," *Synth. Lectures Commun. Netw.*, vol. 12, no. 2, pp. 1–224, Dec. 2019.

[11] A. Kosta, N. Pappas, and V. Angelakis, "Age of information: A new concept, metric, and tool," *Found. Trends Netw.*, vol. 12, no. 3, pp. 162–259, 2017.

[12] H. Chen, Y. Gu, and S.-C. Liew, "Age-of-information dependent random access for massive IoT networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, Jul. 2020, pp. 177–182.

[13] P. D. Mankar, Z. Chen, M. A. Abd-Elmagid, N. Pappas, and H. S. Dhillon, "Throughput and age of information in a cellular-based IoT network," 2020, *arXiv:2005.09547*. [Online]. Available: http://arxiv.org/abs/2005.09547

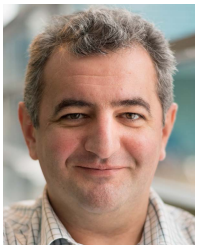
[14] D. Niyato, D. I. Kim, P. Wang, and L. Song, "A novel caching mechanism for Internet of Things (IoT) sensing service with energy harvesting," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[15] N. Pappas, Z. Chen, and M. Hatami, "Average AoI of cached status updates for a process monitored by an energy harvesting sensor," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, USA, Mar. 2020, pp. 1–5.

[16] I. Krikidis, "Average age of information in wireless powered sensor networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 628–631, Apr. 2019.
[17] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger, "Age-optimal constrained cache updating," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 141–145.
[18] M. Bastopcu and S. Ulukus, "Information freshness in cache updating systems," 2020, *arXiv:2004.09475*. [Online]. Available: http://arxiv.org/abs/2004.09475
[19] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Orlando, FL, USA, Jun. 2015, pp. 3008–3012.
[20] X. Wu, J. Yang, and J. Wu, "Optimal status update for age of information minimization with an energy harvesting source," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 1, pp. 193–204, Mar. 2018.
[21] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor, "Using erasure feedback for online timely updating with an energy harvesting sensor," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Orlando, FL, USA, Jul. 2019, pp. 607–611.
[22] A. Arafa and S. Ulukus, "Timely updates in energy harvesting two-hop networks: Offline and online policies," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4017–4030, Jun. 2019.
[23] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor, "Age-minimal transmission for energy harvesting sensors with finite batteries: Online policies," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 534–556, Jan. 2020.
[24] H. Zhu, Y. Cao, X. Wei, W. Wang, T. Jiang, and S. Jin, "Caching transient data for Internet of Things: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2074–2083, Apr. 2019.
[25] S. Leng and A. Yener, "Age of information minimization for wireless ad hoc networks: A deep reinforcement learning approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.
[26] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the Internet of Things," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7468–7482, Nov. 2019.
[27] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, "A reinforcement learning framework for optimizing age of information in RF-powered communication systems," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4747–4760, Aug. 2020.
[28] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," in *Proc. Inf. Theory Appl. Workshop (ITA)*, San Diego, CA, USA, Feb. 2015, pp. 25–31.
[29] C. Tunc and S. Panwar, "Optimal transmission policies for energy harvesting age of information systems with battery recovery," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2019, pp. 2012–2016.
[30] S. Leng and A. Yener, "Age of information minimization for an energy harvesting cognitive radio," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 2, pp. 427–439, Jun. 2019.
[31] G. Stamatakis, N. Pappas, and A. Traganitis, "Control of status updates for energy harvesting devices that monitor processes with alarms," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.
[32] E. T. Ceran, D. Gunduz, and A. Gyorgy, "Reinforcement learning to minimize age of information with an energy harvesting sensor with HARQ and sensing cost," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Paris, France, Apr. 2019, pp. 656–661.
[33] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, and X. Shen, "Deep reinforcement learning for autonomous Internet of Things: Model, applications and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1722–1760, 3rd Quart., 2020.
[34] B. Yin *et al.*, "Only those requested count: Proactive scheduling policies for minimizing effective age-of-information," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr. 2019, pp. 109–117.
[35] F. Li, Y. Sang, Z. Liu, B. Li, H. Wu, and B. Ji, "Waiting but not aging: Optimizing information freshness under the pull model," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 465–478, Feb. 2021.
[36] M. Hatami, M. Jahandideh, M. Leinonen, and M. Codreanu, "Age-aware status update control for energy harvesting IoT sensors via reinforcement learning," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, London, U.K., Aug. 2020, pp. 1–6.
[37] N. Michelusi, K. Stamatiou, and M. Zorzi, "Transmission policies for energy harvesting sensors with time-correlated energy supply," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2988–3001, Jul. 2013.
[38] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[39] Y.-P. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *IEEE Trans. Mobile Comput.*, vol. 19, no. 12, pp. 2903–2915, Dec. 2020.
[40] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queuing network control," *Math. Oper. Res.*, vol. 24, no. 2, pp. 293–305, May 1999.
[41] J. Gittins, K. Glazebrook, and R. Weber, *Multi-Armed Bandit Allocation Indices*. Hoboken, NJ, USA: Wiley, 2011.

Mohammad Hatami received the M.Sc. (Tech.) degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2015. He is currently pursuing the Ph.D. degree with the Centre for Wireless Communications, University of Oulu, Finland. His current research interests include information freshness in wireless networks, caching, machine learning for wireless applications, and system design.

Markus Leinonen (Member, IEEE) received the B.Sc. (Tech.) and M.Sc. (Tech.) degrees in electrical engineering and the D.Sc. (Tech.) degree in communications engineering from the University of Oulu, Finland, in 2010, 2011, and 2018, respectively. In 2010, he joined the Centre for Wireless Communications, University of Oulu, where he is currently working as an Academy of Finland Post-Doctoral Researcher. In 2013, he was a Guest Researcher with the Technical University of Munich, Germany. In 2020, he was a Visiting Post-Doctoral Researcher with the University of California San Diego (UCSD). His research interests include time-critical and sparsity-aware wireless communications.



Marian Codreanu (Member, IEEE) received the M.Sc. degree from the University POLITEHNICA of Bucharest, Romania, in 1998, and the Ph.D. degree from the University of Oulu, Finland, in 2007. In 2008, he was a Visiting Post-Doctoral Researcher with Prof. Ephremides' Group, University of Maryland, College Park, USA. In 2019, he received Marie Skłodowska-Curie Individual Fellowship and joined Linköping University, where he is currently an Associate Professor. He published over 100 journals and conference papers in the areas of wireless communications and networking, statistical signal processing, mathematical optimization, and information theory. His current research focus is on information freshness optimization, sparse signal processing, and machine learning for wireless networking. His thesis was awarded as the Best Doctoral Thesis within the area of all technical sciences in Finland in 2007. In 2013, the Academy of Finland awarded him a five years Academy Research Fellow position. He received his Docent from the Centre for Wireless Communications, University of Oulu, in 2013.