

Minimizing Average On-Demand AoI in an IoT Network with Energy Harvesting Sensors

Mohammad Hatami¹, Markus Leinonen¹, and Marian Codreanu²

¹Centre for Wireless Communications, University of Oulu, Finland

²Department of Science and Technology, Linköping University, Sweden

Email: mohammad.hatami@oulu.fi, markus.leinonen@oulu.fi, marian.codreanu@liu.se

Abstract—Delivering timely status information of a random process has become increasingly important for time-sensitive applications, e.g., vehicle tracking and environment monitoring. We consider an IoT sensing network, where a cache-enabled wireless edge node receives *on-demand* requests from multiple users to send status updates on physical quantities, each measured by an energy harvesting sensor. To serve users' requests, the edge node uses the current information state (i.e., the number of requests, battery level, and AoI for each sensor) to decide whether to command a sensor to send a status update or to retrieve the most recently received sensor's measurements 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, i.e., *average on-demand AoI*. We model this as a Markov decision process problem and derive a relative value iteration algorithm to find an optimal policy. Simulation results illustrate the threshold-based structure of an optimal policy and show that the proposed on-demand updating policy outperforms the greedy (myopic) policy and also, by accounting for the per-sensor request frequencies and intensities, the pure average AoI minimization policy that keeps the edge node updated regardless of requests.

Index Terms—Age of information (AoI), energy harvesting (EH), relative value iteration algorithm (RVIA).

I. INTRODUCTION

Energy harvesting (EH) sensors is one of the key elements for the next generation Internet of Things (IoT) networks [1]. A typical IoT sensing network consists of multiple sensors that measure physical quantities and send them to a destination for further processing. Two inherent features of such networks are: 1) stringent energy limitations of battery-powered sensors which, however, may be counteracted by *harvesting* energy from environmental sources such as sun, heat, and RF ambient, and 2) *transient* nature of data, i.e., the sensors' measurements become outdated after a while.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 send a minimal number of measurements to conserve the energy while providing the users highly fresh data, as required by time-sensitive applications. The freshness of information from the users' perspective can be quantified by the recently emerged metric, the *age of information* (AoI) [2], [3].

In this paper, we consider an IoT status update system consisting of multiple users, multiple EH IoT sensors, and an edge node. The users receive time-sensitive information about physical quantities, each measured by a sensor. The users send requests to the edge node (a gateway) which has a cache storage to store the most recently received measurements

of each physical quantity. To serve a user's request, the edge node 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. Thus, there is an inherent *trade-off* between the AoI at the users and conservation of the sensors' energy in the finite batteries.

Contributions: We derive a Markov decision process (MDP) model for the problem of finding an *optimal policy*, i.e., optimal actions of the edge node at each slot, that minimizes the average *on-demand AoI*, i.e., the average AoI about the physical quantities at the users restricted to the users' request moments. 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. We calculate the state transition probabilities and propose a relative value iteration algorithm (RVIA) that obtains an optimal policy. Numerical experiments illustrate the threshold-based structure of an optimal policy and show that the proposed on-demand updating policy outperforms the greedy policy and also the pure average AoI minimization policy that keeps the edge node updated regardless of requests.

Related works: In [4], the authors proposed MDP-based AoI-optimal scheduling algorithms for a wireless broadcast network where a base station is updating the users on random information arrivals under a transmission capacity constraint. In [5], the authors proposed an online deep reinforcement learning algorithm that minimizes the weighted average AoI plus energy cost for cache updating in an IoT network under dynamic content popularities. The above works do not consider EH sources. In [6], the authors analyzed the average AoI in a cache enabled status updating system with an EH sensor. In [7], [8], the authors derived age-optimal policies for a continuous-time single EH sensor. In [9], 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. In [10], the authors studied AoI minimization in cognitive radio EH communications. In [11], the authors studied age-optimal policies for an EH device that monitors a stochastic process that switches between a normal and alarm state. In [12], the authors studied age-optimal policies for cases where the channel and EH statistics are either known or unknown. Different from [10]–[12], we investigate multiple EH sensors sending updates to the users through a network controller (i.e., the edge node).

In [13], [14], we introduced a concept of on-demand AoI and developed MDP-based learning methods to find optimal policies that minimize the average on-demand AoI in an IoT EH network. Freshness on demand was recently considered in [15], where the authors introduced the average AoI at query (QAoI) and developed an MDP-based policy iteration method to find an optimal policy that minimizes the average QAoI for a sensor that is queried to transmit updates to an edge node over a wireless channel under the sensor's energy constraint. The QAoI metric [15] is equivalent to our on-demand AoI when particularized to the single-user single-sensor case. Here, we extend the on-demand AoI [13], [14] to account for the number of requests for each physical quantity at each time slot, and we incorporate the request information as a part of the state to improve the average on-demand AoI.

II. SYSTEM MODEL

A. Network Model

We consider an IoT sensing network consisting of a set $K = \{1, \dots, K\}$ of K energy harvesting (EH) sensors, an edge node, and a set $N = \{1, \dots, N\}$ of N users, as depicted in Fig. 1. Users receive information about physical quantities (e.g., temperature or humidity) which are independently measured by the K sensors; sensor $k \in K$ measures a physical quantity f_k . We assume that there is no direct link between the users and the sensors, i.e., the users' requests for the values of f_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. Let $r_{k,n}(t) \in \{0, 1\}$, $t = 1, 2, \dots$ denote the random process of requesting the value of f_k by user n ; $r_{k,n}(t) = 1$ if the value of f_k is requested by user $n \in N$ at slot t and $r_{k,n}(t) = 0$ otherwise. The requests are independent across the users, sensors, and time slots. Let $p_{k,n}$ be the probability that the value of f_k is requested by user n at each time slot, i.e., $\Pr\{r_{k,n}(t) = 1\} = p_{k,n}$.

The edge node is equipped with a cache that stores the most recently received *status update packet*¹ from each sensor. Upon receiving a request for the value of f_k at slot t , the edge node can either command sensor k to send a status update or use the previous measurement from the cache, to serve the request. Let $a_k(t) \in \{0, 1\}$ be the *command* action of the edge node at slot t ; $a_k(t) = 1$ if the edge node commands sensor k to send an 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 . We also assume that the sensors have independent communication links to the edge node. Accordingly, at each slot t , the command actions $a_k(t)$, $k \in K$, are independent across k . Note that while the communications between the edge node and the users are

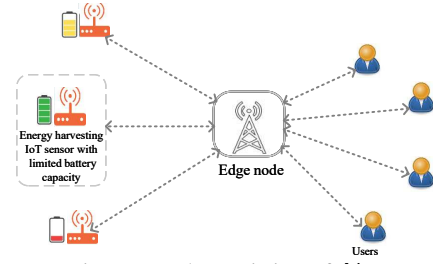


Fig. 1: An IoT sensing network consisting of N users, an edge node (i.e., the gateway), and K energy harvesting sensors.

assumed to be error-free, the transmissions from the sensors to the edge node are prone to errors², as detailed in Section II-A2.

1) *Energy Harvesting Sensors*: 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). Let $b_k(t)$ denote the battery level of sensor k at the beginning of slot t . Thus, $b_k(t) \in \{0, \dots, B_k\}$.

We consider a common assumption (see e.g., [8]) 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., $a_k(t) = 1$), the sensor sends a status update if it has at least one unit of energy, 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 and $d_k(t) = 0$ otherwise. Accordingly, we have

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

where $\mathbb{1}_{\{ \cdot \}}$ is the indicator function. Note that $d_k(t)$ 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 , $k \in K$. Let $e_k(t) \in \{0, 1\}$, $t = 1, 2, \dots$, be 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., $\Pr\{e_k(t) = 1\} = \lambda_k$, $\forall t$.

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

$$b_k(t+1) = \min\{b_k(t) + e_k(t) - d_k(t), B_k\} \quad (2)$$

2) *Links Between the Edge Node and the Sensors*: We consider an *error-free* binary/single-bit *command* link from the edge node to each sensor, and an *error-prone* wireless link from each sensor to the edge node, i.e., a transmission through the link is either *successful* or *failed*. Let $h_k(t) \in \{0, 1\}$ denote the event that a status update from sensor k is successfully received by the edge node at slot t . Otherwise, $h_k(t) = 0$, which accounts for both that 1) sensor k sends a status update but the transmission is failed, or 2) the sensor does not send an update. Let β_k be the conditional probability that, given that sensor k transmits a status update, it is successfully received by the edge node, i.e., $\Pr\{h_k(t) = 1 | d_k(t) = 1\} = \beta_k$, $\forall t$, which represents the *transmit success probability* of the link from sensor k to the edge node.

3) *Age of Information (AoI)*: *AoI* is a destination-centric metric that quantifies the freshness of information of a re-

¹A status update packet contains the measured value of a monitored process and a time stamp representing the time when the sample was generated.

²Typically, the edge node accesses to sufficient power whereas the sensors rely only on the energy harvested from the environment.

motely observed random process [2]. Formally, let $f_k(t)$ 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 slot in which the edge node received a status update packet from sensor k , i.e., $u_k(t) = \max_{t^0 < t} h_k(t^0) = 1$; thus, the AoI about f_k is written as $f_k(t) = t - u_k(t)$. We make a common assumption (see e.g., [9]–[15]) that $f_k(t)$ is upper-bounded by a finite value $f_{k,\max}$, i.e., $f_k(t) \leq f_{k,\max}$; $2; \dots; k, \max g$. This is reasonable, because once $f_k(t)$ reaches a high value $f_{k,\max}$, the available measurement about quantity 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 $f_k(t)$ can be written as

$$f_k(t+1) = \begin{cases} 1; & \text{if } h_k(t) = 1; \\ \min\{f_k(t) + 1; f_{k,\max}\}; & \text{if } h_k(t) = 0; \end{cases} \quad (3)$$

B. State Space, Action Space, and Cost Function

1) *State*: Let $s_k(t) \in S_k$ denote the state associated with sensor k at slot t , which is defined as $s_k(t) = (r_k(t); b_k(t); f_k(t)g)$, where 1) $r_k(t) = \sum_{n=1}^N r_{k;n}(t) \in \{0; 1; \dots; N\}g$ indicates how many users requested for f_k (i.e., the number of requests for f_k at slot t), 2) $b_k(t) \in \{0; 1; \dots; B_k\}g$ is the battery level, and 3) $f_k(t) \in \{1; 2; \dots; f_{k,\max}\}g$ is the AoI about f_k at the edge node; S_k is the per-sensor state space with dimension $|S_k| = (N+1)(B_k+1) f_{k,\max}$. Thus, the state of the system at slot t is expressed as $\mathbf{s}(t) = (s_1(t); \dots; s_K(t)) \in S$, $S = S_1 \times \dots \times S_K$.

2) *Action*: As discussed in Section II-A, at each time slot the edge node decides whether to command sensor k to send a status update (and update the cache) or not, i.e., $a_k(t) \in A_k = \{0; 1\}g$, where A_k is the per-sensor action space. Thus, the action of the edge node at slot t is expressed as $\mathbf{a}(t) = (a_1(t); \dots; a_K(t)) \in A$, $A = A_1 \times \dots \times A_K$.

3) *Immediate Cost Function*: We consider a cost function that penalizes the information staleness of the requested measurements received by the users. Accordingly, we define the per-sensor immediate cost at slot t as the *on-demand AoI* as

$$c_k(t) = \sum_{n=1}^N r_{k;n}(t) f_k(t+1) = r_k(t) f_k(t+1); \quad (4)$$

Note that when the value of f_k is not requested by any user 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 by the end of the same slot, $f_k(t+1)$ is the effective AoI about f_k seen by the users. Note that the (per-sensor) immediate cost in (4) represents the cost of taking action $a_k(t)$ in state $s_k(t)$, i.e., $c_k(t) = c_k(s_k(t); a_k(t))$.

III. PROBLEM FORMULATION AND OPTIMAL POLICY

A. Problem Formulation

A policy, denoted by π , is a mapping from the state space to the action space, i.e., $\pi: S \rightarrow A$; the action taken in state

\mathbf{s} under policy π is denoted by $\pi(\mathbf{s})$. Under a policy π , the average cost over all the sensors and users is given by

$$C = \lim_{T \rightarrow \infty} \frac{1}{KT} \sum_{t=1}^T \sum_{k=1}^K E [c_k(t) | \mathbf{s}(0)]; \quad (5)$$

where $E[\cdot]$ denotes the expected value of $c_k(t)$ given that the edge node follows policy π , and $\mathbf{s}(0) = (s_1(0); \dots; s_K(0))$ is the initial state. We aim to find an optimal policy π^* that achieves the minimum average cost, i.e., $\pi^* = \arg \min_{\pi} C$.

Remark 1. As described in Section II-A, the command actions $a_k(t)$ at any slot t are independent across sensors $k \in \{1, \dots, K\}$. Thus, the problem of finding an optimal policy that minimizes (5) is *separable* across the sensors; minimizing the cost in (5) reduces to minimizing K per-sensor average costs, defined as

$$C_k = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T E [c_k(t) | s_k(0)]; \quad (6)$$

Thus, our objective is to find optimal per-sensor policies π_k , $\forall k \in \{1, \dots, K\}$, that achieve the minimum average cost for each sensor k , i.e., $\pi_k^* = \arg \min_{\pi_k} C_k$.

Remark 2. For those states in which there are no requests for the value of f_k at slot t (i.e., $s_k(t) = (0; b_k(t); f_k(t)g)$), the optimal action is $\pi_k(s_k(t)) = 0$. In this case, the immediate cost (4) becomes zero (i.e., $c_k(t) = 0$) and the action $a_k(t) = 0$ leads to energy saving for sensor k , which can be used later to serve the users with fresh measurements.

B. MDP Modeling and Optimal Policy

We model the problem of finding an optimal per-sensor policy³ as an MDP and we find such an optimal policy using *relative value iteration algorithm (RVIA)*. The MDP model associated with sensor k is defined by the tuple $(S_k; A_k; c_k; P_k; \gamma)$, where S_k , A_k , and $c_k(s_k(t); a_k(t))$ were defined in Section II-B. The state transition probability $P_k(s_k(t+1) | s_k(t); a_k(t))$ maps a state-action pair at slot t onto a distribution of states at slot $t+1$; the probability of transition from current state $(s_k(t)) = (r; b; g)$ to next state $(s_k(t+1)) = (r^0; b^0; g^0)$ under action $a_k(t) = a$ is given by

$$P_k(r^0; b^0; g^0 | r; b; g; a) = \Pr\{r^0 | r; a\} \Pr\{b^0 | b; a\} \Pr\{g^0 | g; a\}; \quad (7)$$

where

$$\Pr\{r^0\} = \begin{cases} \prod_{n=1}^N (1 - p_{k;n}); & r^0 = 0; \\ \sum_{m \in n} \prod_{n=1}^N p_{k;n} (1 - p_{k;m}); & r^0 = 1; \\ \vdots & \vdots \\ \prod_{n=1}^N p_{k;n}; & r^0 = N; \\ 0; & \text{otherwise.} \end{cases}$$

$$\Pr\{b^0 | b < B_k; a = 0\} = \begin{cases} < k; & b^0 = b + 1; \\ 1 - p_{k;}; & b^0 = b; \\ 0; & \text{otherwise.} \end{cases}$$

$$\Pr\{b^0 | b = B_k; a = 0\} = \begin{cases} 1 - p_{k;}; & b^0 = b; \\ 0; & \text{otherwise.} \end{cases}$$

$$\Pr\{b^0 | b = 0; a = 1\} = \begin{cases} < k; & b^0 = 1; \\ 1 - p_{k;}; & b^0 = 0; \\ 0; & \text{otherwise.} \end{cases}$$

³We present the derivation of such an optimal policy for a particular sensor k ; the edge node runs in parallel one policy for each sensor in the network.

$$\begin{aligned} \Pr(b^j | b = 1; a = 1) &= \begin{cases} 1 & k; \quad b^j = b; \\ 0 & \text{otherwise.} \end{cases} \\ \Pr(b^j | b; a = 0) &= \begin{cases} 1 & \text{if } b^j = \min\{f\Delta + 1, \Delta_{k, \max} g\}; \\ 0 & \text{otherwise.} \end{cases} \\ \Pr(b^j | b = 1; a = 1) &= \begin{cases} 1 & k; \quad b^j = \min\{f + 1, k, \max\{g\}; \\ 0 & \text{otherwise.} \end{cases} \\ \Pr(b^j | b = 0; a = 1) &= \begin{cases} 1 & \text{if } b^j = \min\{f\Delta + 1, \Delta_{k, \max} g\}; \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The following Lemma characterizes an optimal policy π_k that minimizes (6).

Lemma 1. *An optimal policy π_k is obtained by solving the following Bellman's equation [16, Prop. 4.2.1]*

$$C_k + v_k(s) = \min_{a \in \mathcal{A}_k} q_k(s; a); \quad s \in S_k; \quad (8)$$

where $v_k(s)$ is the optimal state-value function, C_k is the optimal average cost achieved by π_k which is independent of the initial state $S_k(0)$, and $q_k(s; a)$ is the optimal action-value function, which is given by

$$q_k(s; a) = \mathbb{P}_{s^0 \in S_k} P_k(s^0 | s; a) [c_k(s; a) + v_k(s^0)]; \quad (9)$$

Then, an optimal action taken in state s is expressed as $\pi_k(s) = \arg \min_{a \in \mathcal{A}_k} q_k(s; a); \quad s \in S_k$.

The optimal state-value function $v_k(s)$, $s \in S_k$, and consequently, an optimal policy π_k , can be computed by turning the Bellman optimality equation (8) into an iterative procedure, RVIA. Particularly, the state-value function in state s at iteration $i = 1; 2; \dots$, denoted by $v_k^{(i)}(s)$, is updated as

$$v_k^{(i)}(s) = \min_{a \in \mathcal{A}_k} q_k^{(i)}(s; a) = v_k^{(i-1)}(s_{\text{ref}}); \quad s \in S_k; \quad (10)$$

where $q_k^{(i)}(s; a) = \mathbb{P}_{s^0 \in S_k} P_k(s^0 | s; a) [c_k(s; a) + v_k^{(i-1)}(s^0)]$ and $s_{\text{ref}} \in S_k$ is an arbitrary reference state. For any arbitrary initialization $v_k^{(0)}(s)$, the sequence $\{v_k^{(i)}(s)\}_i$ converges to the optimal state-value function $v_k(s)$, i.e., $\lim_{i \rightarrow \infty} v_k^{(i)}(s) = v_k(s)$, $\forall s \in S_k$, [16, Section 4.3]. The proposed RVIA is presented in Algorithm 1.

IV. SIMULATION RESULTS

The simulation scenario consists of $N = 3$ users, i.e., $N = \{1; 2; 3\}$, and $K = 4$ sensors, i.e., $K = \{1; 2; 3; 4\}$, with battery capacity $B_k = 15; \forall k \in K$, units of energy.

Fig. 2 illustrates the structure of an optimal policy for sensor 1, i.e., π_1 . Each point represents a potential per-sensor state as a three-tuple $s = (r; b; g)$. For each such state, a red circle indicates that the optimal action is that the edge node does not command the sensor (i.e., $\pi_1(s) = 0$), whereas a blue square indicates that the optimal action is to command the sensor (i.e., $\pi_1(s) = 1$). The set of the blue points is referred to as the *command region* hereinafter.

From Figs. 2(a)–(d), it can be concluded that an optimal policy for each sensor has a *threshold-based* structure with respect to the number of requests r , battery level b , and AoI g . For instance, consider state $s = (1; 5; 50)$ in which $\pi_1(s) = 1$; then, by the threshold-based structure, $\pi_1(\underline{s}) = 1$ for all states $\underline{s} = (r; b; g)$, $r \leq 1$, $b \leq 5$, $g \leq 50$. In particular, since the cost function (4) is linearly increasing with $r_k(t)$,

Algorithm 1 RVIA that obtains optimal policies $\pi_k, \forall k$.

```

1: Initialize  $v_k(s) = 0, k \in K; \forall s \in S_k$ , determine a reference
   state  $s_{\text{ref}} \in S_k$  and a small threshold  $\epsilon > 0$ .
2: for  $k = 1; \dots; K$  do
3:   repeat
4:      $v_{\text{tmp}}(s) = \min_{a \in \mathcal{A}_k} \sum_{s^0 \in S_k} P_k(s^0 | s; a) [c_k(s; a) + v_k(s^0)]$ 
        $v_k(s_{\text{ref}})$ , for all  $s \in S_k$ .
5:      $v_k(s) = \max_{s^0 \in S_k} (v_{\text{tmp}}(s) - v_k(s^0)) \min_{s \in S_k} (v_{\text{tmp}}(s) - v_k(s))$ .
6:      $v_k(s) = v_{\text{tmp}}(s)$ , for all  $s \in S_k$ .
7:   until  $\epsilon < \max_{s \in S_k} |v_k(s) - v_{\text{tmp}}(s)|$ 
8:    $\pi_k(s) = \arg \min_{a \in \mathcal{A}_k} \sum_{s^0 \in S_k} P_k(s^0 | s; a) [c_k(s; a) +$ 
        $v_k(s^0)]$ , for all  $s \in S_k$ .
9: end for

```

the edge node has more incentive to command a sensor that is associated with a large number of requests. Furthermore, as shown in Fig. 2(a), if there are no requests for r_1 (i.e., $r = 0$), the optimal action is that the edge node does not command the sensor, regardless of the battery level and AoI, i.e., $\pi_1(0; b; g) = 0$, as discussed in Remark 2.

Fig. 3 depicts the optimal action in each state $s = \{2; b; g\}$ for each sensor. By comparing Fig. 3(a) and Fig. 3(c), it is inferred that the command region enlarges by increasing the EH rate ($\rho_1 < \rho_3$, $\rho_1 = \rho_3$, and $\rho_{1:n} = \rho_{3:n}; \forall n$). This is because when a sensor harvests energy more often, it can send updates more often. Fig. 3(a) and Fig. 3(d) show that the command region enlarges also by increasing the transmit success probability ($\rho_1 = \rho_4$, $\rho_1 > \rho_4$, and $\rho_{1:n} = \rho_{4:n}; \forall n$). Namely, when the link becomes more reliable, the edge node commands the sensor more often as it is more confident of receiving the transmitted update packet. By comparing Fig. 3(a) and Fig. 3(b), it is deduced that when the sensor is requested more often, the command region shrinks ($\rho_1 = \rho_2$, $\rho_1 = \rho_2$, and $\rho_{1:n} < \rho_{2:n}; \forall n$), i.e., the edge node commands the sensor less to save its energy for the future requests.

Fig. 4 depicts the performance of the proposed RVIA. In the *greedy (myopic)* policy, the edge node commands sensor k whenever the value of r_k is requested by at least one user (i.e., $r_k(t) \geq 1$). Furthermore, to highlight the improvement (in terms of reducing the average cost) obtained by incorporating the request information as a part of the state, we also consider a method in which the edge node commands based only on the battery level and the AoI, i.e., $S_k(t) = \{b_k(t); g_k(t)\}$, referred to as *pure AoI* hereinafter. In other words, the pure AoI policy minimizes the average AoI at the edge node, regardless of requests. As illustrated, the RVIA reduces the average cost approximately 30 % compared to the greedy (myopic) policy and roughly 10 % compared to the pure AoI method.

V. CONCLUSIONS

We investigated on-demand status updating in an IoT sensing network consisting of multiple users, multiple EH sensors, and a wireless edge node. We modeled the problem as an MDP and found an optimal policy that minimizes the average on-demand AoI. We numerically depicted that optimal policies have threshold-based structures and showed that the

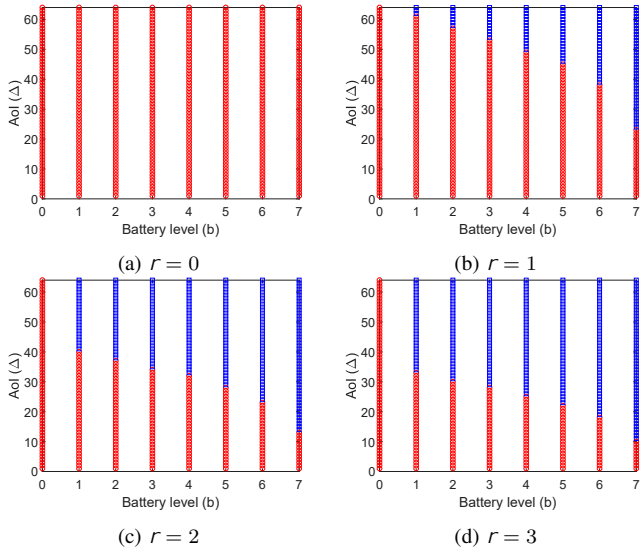


Fig. 2: Structure of an optimal policy for sensor 1 (i.e., γ_1) for each state $s = fr; b; g$, where $\gamma_1 = 0.02$, $\gamma_1 = 0.85$, and $p_{1;n} = 0.05; 8n \geq N$. Red circle: no command $a = 0$; blue square: command $a = 1$.

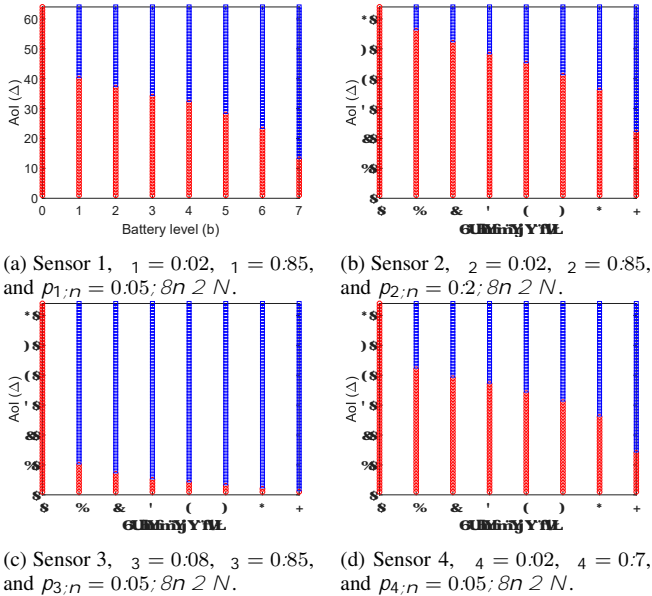


Fig. 3: Optimal policies for each sensor in states $s = fr; b; g$.

proposed on-demand updating policy outperforms the greedy (myopic) policy and also, by exploiting the per-sensor request frequencies and intensities, the pure average AoI minimization policy.

VI. ACKNOWLEDGMENTS

This research has been financially supported by the Infotech Oulu, the Academy of Finland (grant 323698), and Academy of Finland 6Genesis Flagship (grant 318927). The work of M. Leinonen has also been financially supported in part by the Academy of Finland (grant 319485). M. Hatami would like to acknowledge the support of HPY Research Foundation and Riitta ja Jorma J. Takanen Foundation.

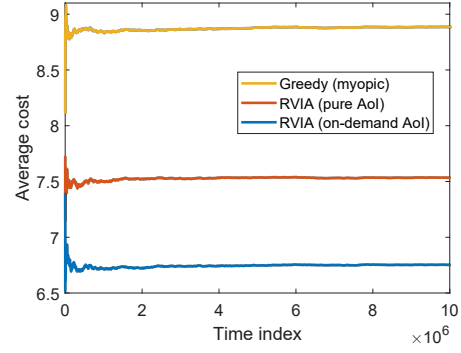


Fig. 4: Performance of the proposed RVIA algorithm.

REFERENCES

- [1] F. Tariq, M. R. A. Khandaker, K.-K. Wong, M. A. Imran, M. Bennis, and M. Debbah, "A speculative study on 6G," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 118–125, 2020.
- [2] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE Int. Conf. on Computer. Commun. (INFOCOM)*, Orlando, FL, USA, Mar. 25–30, 2012, pp. 2731–2735.
- [3] Y. Sun, I. Kadota, R. Talak, E. Modiano, and R. Srikant, "Age of information: A new metric for information freshness," *Synthesis Lectures on Communication Networks*, vol. 12, no. 2, pp. 1–224, 2019.
- [4] 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, 2020.
- [5] X. Wu, X. Li, J. Li, P. C. Ching, and H. V. Poor, "Deep reinforcement learning for IoT networks: Age of information and energy cost tradeoff," in *Proc. IEEE Global Telecommun. Conf.*, Taipei, Taiwan, Dec. 7–11, 2020, pp. 1–6.
- [6] N. Pappas, Z. Chen, and M. Hatami, "Average AoI of cached status updates for a process monitored by an energy harvesting sensor," in *Proc. Conf. Inform. Sciences Syst. (CISS)*, Princeton, NJ, USA, Mar. 18–20, 2020, pp. 1–5.
- [7] 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. Inform. Theory*, Orlando, FL, USA, Jul. 7–12, 2019, pp. 607–611.
- [8] —, "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.
- [9] 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.
- [10] S. Leng and A. Yener, "Age of information minimization for an energy harvesting cognitive radio," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 5, no. 2, pp. 427–439, May 2019.
- [11] G. Stamatakis, N. Pappas, and A. Traganitis, "Control of status updates for energy harvesting devices that monitor processes with alarms," in *IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, USA, Dec. 9–13, 2019, pp. 1–6.
- [12] E. T. Ceran, D. Gündüz, and A. Gyögy, "Reinforcement learning to minimize age of information with an energy harvesting sensor with HARQ and sensing cost," in *Proc. IEEE INFOCOM Workshop*, Paris, France, Apr. 29–May 2 2019, pp. 656–661.
- [13] 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 Int. Symp. Pers., Indoor, Mobile Radio Commun.*, London, UK, Aug. 31–Sep. 3 2020.
- [14] M. Hatami, M. Leinonen, and M. Codreanu, "AoI minimization in status update control with energy harvesting sensors," *arXiv preprint arXiv:2009.04224*, 2020.
- [15] J. Holm, A. E. Kalør, F. Chiariotti, B. Soret, S. K. Jensen, T. B. Pedersen, and P. Popovski, "Freshness on demand: Optimizing age of information for the query process," *arXiv preprint arXiv:2011.00917*, 2020.
- [16] D. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II*, 3rd ed. Athena Scientific, 2007.