Reinforcement Learning for Minimizing Age of Information under Realistic Physical Dynamics

Sihua Wang*, Mingzhe Chen†,‡, Walid Saad§, Changchuan Yin*, Shuguang Cui‡, and H. Vincent Poor†
*Beijing Laboratory of Advanced Information Network, Beijing University of Posts and Telecommunications, Beijing, China
†Department of Electrical Engineering, Princeton University, Princeton, NJ, USA
‡Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, China
§Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA
Emails: sihuawang@bupt.edu.cn, mingzhec@princeton.edu, walids@vt.edu, ccyin@bupt.edu.cn, shuguangcui@cuhk.edu.cn, poor@princeton.edu.

***Abstract*—In this paper, the problem of minimizing the weighted sum of age of information (AoI) and total energy consumption of Internet of Things (IoT) devices is studied. In particular, each IoT device monitors a physical process that follows nonlinear dynamics. As the dynamic of the physical process varies over time, each device must sample the real-time status of the physical system and send the status information to a base station (BS) so as to monitor the physical process. The dynamics of the realistic physical process will influence the sampling frequency and status update scheme of each device. In particular, as the physical process varies rapidly, the sampling frequency of each device must be increased to capture these physical dynamics. Meanwhile, changes in the sampling frequency will also impact the energy usage of the device. Thus, it is necessary to determine a subset of devices to sample the physical process at each time slot so as to accurately monitor the dynamics of the physical process using minimum energy. This problem is formulated as an optimization problem whose goal is to minimize the weighted sum of AoI and total device energy consumption. To solve this problem, a machine learning framework based on the repeated update Q-learning (RUQL) algorithm is proposed. The proposed method enables the BS to overcome the biased action selection problem (e.g., an agent always takes a subset of actions while ignoring other actions), and hence, dynamically and quickly finding a device sampling and status update policy so as to minimize the sum of AoI and energy consumption of all devices. Simulations with real data of PM 2.5 pollution in Beijing from the Center for Statistical Science at Peking University show that the proposed algorithm can reduce the sum of AoI by up to 26.9% compared to the conventional Q-learning method.**

***Index Terms*—Internet of things, adaptive sampling frequency, age of information, reinforcement learning.**

I. Introduction

The Internet of Things (IoT) will be a key enabler of various cyber-physical systems and networked monitoring applications [1], such as environment monitoring and vehicle tracking. For these IoT applications, the freshness of the status information of the physical process at the operation devices is of fundamental importance for accurate monitoring and control. To quantify the freshness of the status information of the physical process, the age of information (AoI) has recently been proposed as a useful performance metric [2]. The AoI is defined as the duration between the current time and the generation time of the most recently received status update. Compared to conventional delay metrics that measure queuing or transmission latency, AoI considers the generation time of each packet, thus, characterizing the freshness of the status information from the perspective of the destination. Therefore, optimizing the AoI in IoT leads to distinctively different system designs from those used for conventional delay optimization.

A number of existing works have studied important problems related to AoI such as in [3]–[6]. In [3], the authors optimized the AoI of each IoT device for both first-come first-serve and last-come first-serve systems. The authors in [4] proposed optimal status update schemes for an energy harvesting source to minimize the average AoI of IoT devices. The sum AoI of IoT devices is minimized under throughput constraints in [5]. The authors in [6] introduced an optimal status update scheme to minimize the average AoI of all devices. However, the existing works in [3]–[6] only investigated the optimization of the sampling policy without considering the dynamics of the physical process. In fact, the dynamics of a realistic physical process will strongly influence the optimization of the status sampling and updating schemes. For example, as the physical process varies rapidly, an IoT device must increase the sampling frequency so as to capture these physical dynamics. In contrast, the IoT device can save energy by reducing its sampling frequency when the physical process is varying slowly. Therefore, it is necessary to analyze the dynamics of the realistic physical process so as to dynamically adjust the sampling frequency and optimize the status sampling and updating schemes. However, the physical process dynamics are time-dependent, and hence, a Markov decision process that satisfies the non-aftereffect property as done in [3]–[6] cannot be used to optimize the sampling policy for the physical process.

The main contribution of this paper is a novel framework that enables a BS to adapt the sampling policy for IoT devices so as to dynamically minimize AoI and device energy consumption. In particular, we study a real-time IoT system, in which each device is used to sample the status of a realistic physical process that is modeled using a nonlinear dynamical equation. The relationship between the dynamics of the physical process and the sampling frequency of the device is analyzed to enable the BS to optimize the sampling policy of each device and capture the variation of the physical process. This problem is formulated as an optimization problem whose goal is to minimize the weighted sum of AoI and energy consumption of all devices. To solve this problem, a repeated update Q-learning (RUQL) algorithm is proposed to optimize the sampling policy.

The work was supported in part by Beijing Natural Science Foundation and Municipal Education Committee Joint Funding Project under Grant KZ201911232046, the National Natural Science Foundation of China under Grants 61671086 and 61871041, by Beijing Laboratory Funding under Grant 2019BJLAB01, by the 111 Project under Grant B17007, by the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, by Natural Science Foundation of China with grant NSFC-61629101, by the U.S. National Science Foundation under Grants CNS-1739642, CCF-0939370, and CCF-1908308, and in part by BUPT Excellent Ph.D. Students Foundation (CX2020307).

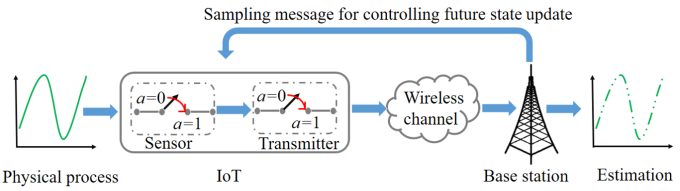

Fig. 1. An illustration of the considered IoT network.

Compared to a traditional reinforcement learning (RL) algorithm with a fixed learning rate, the proposed method enables the BS to adjust the learning rate based on each specific action in order to avoid the policy-bias problem. Simulations with real data of PM 2.5 pollution in Beijing from the Center for Statistical Science at Peking University show that the proposed algorithm can reduce the sum of AoI by up to 26.9% compared to the conventional Q-learning (QL) method. *To the best of our knowledge, this is the first work that considers the optimization of the sampling policy for a real-time IoT system consisting of a realistic physical process.*

II. System Model and Problem Formulation

Consider a real-time IoT system that consists of a BS and a set $\mathcal{M}$ of M IoT devices. In the studied model, each IoT device is equipped with a sensor that monitors the real-time status of a physical system (e.g., an atmospheric sampler that monitors the variation of the atmospheric environment) and a transmitter that sends the monitored information to the BS through a wireless channel. Here, the process of monitoring the physical system is called *status sampling* while the process of sending the sampled information to the BS is called *status update*. After receiving the status information related to the state of the physical system, the BS selects a subset of IoT devices to sample the physical process at next time slot, as illustrated in Fig. 1. Next, we first introduce the model of the physical process. Then, we explain the AoI model to measure the freshness of status information of the physical process at IoT device and BS, respectively.

A. Model of Physical Process

We consider heterogeneous nonlinear time-varying dynamics to describe the variation of the physical process monitored by the IoT devices. These heterogeneous nonlinear dynamics of the physical process over discrete time t can be written as [7]

$$\boldsymbol{x}_{m,t+1} = \boldsymbol{A}_m \boldsymbol{x}_{m,t} + \boldsymbol{B}_m f_m(\boldsymbol{x}_{m,t}) + \boldsymbol{\epsilon}_{m,t}, \quad (1)$$

where $\boldsymbol{x}_{m,t} \in \mathbb{R}^{Z_m}$ is the system state vector sampled by device m at time slot t with Z_m representing the data size of status information of device m and $\boldsymbol{\epsilon}_{m,t}$ is a random process independent of the system state. $f_m(\cdot) : \mathbb{R}^{Z_m} \to \mathbb{R}^{Z_m}$ is a nonlinear function satisfying $f_m(\mathbf{0}) = \mathbf{0}$. $\boldsymbol{A}_m$ and $\boldsymbol{B}_m$ are constant matrices. Note that, (1) has been widely used to model the physical process of nonlinear dynamic systems such as wide-area irrigation systems, electric power grids, automated highway systems, and environmental detection systems. For example, the dynamics of the atmospheric environment quality can be captured by (1) with $\boldsymbol{x}_{m,t}$ being the current air pollution index and $\boldsymbol{x}_{m,t+1}$ being the dynamics of the air pollution index while $\boldsymbol{A}_m \boldsymbol{x}_{m,t}$ and $\boldsymbol{B}_m f_m(\boldsymbol{x}_{m,t})$ represent the linear and nonlinear function to capture the effects of wind and precipitation. Using (1), the current system state can be estimated based on the latest observed state, which is given by [8]

$$\hat{\boldsymbol{x}}_{m,t} = \boldsymbol{A}_m^{\delta(t)} \boldsymbol{x}_{m,t-\delta(t)} + \sum_{q=1}^{\delta(t)} \boldsymbol{A}_m^{q-1} \boldsymbol{B}_m f_m(\boldsymbol{x}_{m,t-q}), \quad (2)$$

where $\boldsymbol{x}_{m,t-\delta(t)}$ is the latest status information generated at time slot $t-\delta(t)$ with $\delta(t)$ being the duration of the generation time between $\boldsymbol{x}_{m,t}$ and $\boldsymbol{x}_{m,t-\delta(t)}$. Given the estimation of the system state vector at time slot t, the state estimation error can be expressed as

$$\boldsymbol{y}_{m,t} = \hat{\boldsymbol{x}}_{m,t} - \boldsymbol{x}_{m,t}. \quad (3)$$

In fact, $\boldsymbol{y}_{m,t}$ measures the estimation error of the dynamics and can be used to determine the sampling frequency of device m at each time slot. To obtain the sampling frequency, we first need to calculate the maximum variation frequency of the physical process by analyzing the nonlinear dynamics of the physical system. For this purpose, (3) can be linearly approximated by [9]

$$\frac{d\boldsymbol{y}_{m,t}}{dt} = \boldsymbol{J}_{f_m}(\boldsymbol{x}_{m,t}) \cdot \boldsymbol{y}_{m,t} + o(\|\boldsymbol{y}_{m,t}\|), \quad (4)$$

where $\boldsymbol{J}_{f_m}(\boldsymbol{x}_{m,t}) \cdot \boldsymbol{y}_{m,t}$ is the first-order approximation with $\boldsymbol{J}_{f_m}(\boldsymbol{x}_{m,t})$ being the Jacobian matrix of function f_m and $o(\|\boldsymbol{y}_{m,t}\|)$ a high-order approximation that can be neglected compared to $\boldsymbol{J}_{f_m}(\boldsymbol{x}_{m,t}) \cdot \boldsymbol{y}_{m,t}$. Then, we diagonalize $\boldsymbol{J}_{f_m}(\boldsymbol{x}_{m,t})$ to obtain the maximum variation frequency of the physical process at time slot t with respect to $\boldsymbol{J}_{f_m}(\boldsymbol{x}_{m,t})$, which is given by

$$\boldsymbol{J}_{f_m}(\boldsymbol{x}_{m,t}) = \boldsymbol{U} \cdot \mathrm{diag}(\mu_{1,t}, \cdots, \mu_{Z_m,t}) \cdot \boldsymbol{U}^{-1}, \quad (5)$$

where $\mathrm{diag}(\mu_{1,t}, \cdots, \mu_{Z_m,t})$ is a diagonal matrix with $(\mu_{1,t}, \cdots, \mu_{Z_m,t})$ being the eigenvalues of the Jacobian matrix and $\boldsymbol{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{Z_m}]$ is a non-singular matrix with $\boldsymbol{u}_{z_m} \in \mathbb{R}^{Z_m}$ being the corresponding eigenvectors of $\boldsymbol{J}_{f_m}(\boldsymbol{x}_{m,t})$. Based on (5), the time-domain maximum variation frequency of the physical process can be computed as [10]

$$\Omega_{m,t} = \max_{z_{m,t} \in \mathcal{Z}_{m,t}} \left|\mathrm{Im}[\mu_{z_{m,t}}]\right| + \sqrt{\frac{\|\boldsymbol{y}_{m,t}\|_2}{\xi_m^2} - \min_{z_{m,t} \in \mathcal{Z}_{m,t}} \mathrm{Re}[\mu_{z_{m,t}}]^2}, \quad (6)$$

where ξ_m is a minimum frequency that device m can distinguish, $\mathrm{Im}[\mu_{z_{m,t}}]$ and $\mathrm{Re}[\mu_{z_{m,t}}]$ is the imaginary part and real part of $\mu_{z_{m,t}}$, respectively. By assigning the sampling frequency $F_{m,t} = \Omega_{m,t}/\pi$ based on Nyquist theory, the maximum sampling interval of the dynamic physical process $\Delta_{m,t}$ can be written as

$$\Delta_{m,t} = 1/F_{m,t} = \pi/\Omega_{m,t}. \quad (7)$$

From (6) and (7), we can see that, the maximum sampling interval $\Delta_{m,t}$ depends on the state estimation error $\boldsymbol{y}_{m,t}$. As $\boldsymbol{y}_{m,t}$ increases, the maximum variation frequency $\Omega_{m,t}$ increases and hence, $\Delta_{m,t}$ decreases. This is because, as the state estimation error increases, (1) cannot describe the physical process accurately, thus, device m must increase the sampling frequency so as to collect more status information to capture the variation of the physical process and correct (1).

B. AoI Model of IoT Device

Different from existing literature [3]–[6] where the AoI at each device m only depends on the time interval $\delta(t)$ between the current sampling status and the latest historical sampling status, in this model, we consider the dynamic sampling frequency of a real-time physical system and hence, the AoI at

each device m will be affected by the maximum sampling interval $\Delta_{m,t}$ and $\delta(t)$, which is given by

$$\phi_{m,t}(a_{m,t}) = \begin{cases} \max\{0, \delta(t) - \Delta_{m,t}\}, & \text{if } a_{m,t} = 1, \\ \min\{\phi_{m,t-1} + \tau, \phi_{\max}\}, & \text{otherwise}, \end{cases} \quad (8)$$

where τ is the duration of each time slot and $a_{m,t} \in \{0, 1\}$ is the sampling action of device m at time slot t with $a_{m,t} = 1$ indicating that device m samples the physical process and generates a new status packet $\boldsymbol{x}_{m,t}$ at time slot t, and $a_{m,t} = 0$, otherwise. $\phi_{\max}$ is the upper limit of the maximum sampling interval to generate that the value of $\phi_{m,t}(a_{m,t})$ is finite. This is due to that, for time-critical IoT applications, it is not meaningful for the destination node to receive a status information with an infinite age. Such highly outdated status information will not be of any use to the system or underlying application. From (8), we can see that, if device m samples a status at time slot t and the time interval $\delta(t)$ between the current sampling status (at time slot t) and the latest historical sampling status (at time slot $t - \delta(t)$) is less than the maximal sampling interval $\Delta_{m,t}$, then the AoI at device m decreases to zero. This implies that, if $\delta(t)$ is smaller than $\Delta_{m,t}$, the sampling frequency of device m will satisfy the constraint of Nyquist theory and thus, the sampled status can be used to capture the variation of the dynamic physical process accurately. Otherwise, the AoI at device m decreases to $\delta(t) - \Delta_{m,t}$ which implies the latency of the sampling action beyond the maximum sampling interval. On the other hand, if device m does not sample at time slot t, the AoI at device m is increased by τ.

C. AoI Model of Base Station

After generating a status packet at time slot t, device m must transmit the status packet to the BS immediately. We assume that the BS adopts an orthogonal frequency division multiple access (OFDMA) transmission scheme. Let $\mathcal{I}$ be the set of $I \leqslant M$ uplink orthogonal resource blocks (RBs). The BS must select a subset of IoT devices to sample the physical process under the constraint that each RB can be allocated to at most one device. The data rate of the uplink transmission between device m and the BS over each RB is given by (in bits/s) [11]

$$r_{m,t}(a_{m,t}) = a_{m,t} W \log_2 \left(1 + \frac{P_T h_{m,t}}{\sigma_N^2}\right), \quad (9)$$

where W is the RB bandwidth and P_T is the transmit power of each device m. $h_{m,t} = g_{m,t} d_m^{-\beta}$ is the channel gain between device m and the BS where $g_{m,t}$ is a Rayleigh fading channel gain, d_m is the distance between user m and the BS, and β is the path loss exponent. σ_N^2 represents variance of the additive white Gaussian noise. Here, we assume that each device can only occupy one RB for status packet transmission over uplink and hence, $\sum_{m=1}^{M} a_{m,t} \leqslant I$. Based on (9), the uplink transmission delay between device m and the BS is given by

$$l_{m,t}(a_{m,t}) = \frac{Z_m}{r_{m,t}(a_{m,t})}. \quad (10)$$

Given the uplink transmission delay, the AoI at the BS for device m can be determined, which is given by

$$\Phi_{m,t}(a_{m,t}) = \begin{cases} \phi_{m,t}(a_{m,t}) + l_{m,t}(a_{m,t}), & \text{if } a_{m,t} = 1, \\ \Phi_{m,t-1} + \tau, & \text{otherwise}. \end{cases} \quad (11)$$

From (11), we can see that, if device m sends the status packet to the BS at time slot t, then the AoI at BS will be updated to $\phi_{m,t} + l_{m,t}$ otherwise, the AoI increases by τ. For broadcasting the sampling message over downlink subcarriers, we assume that the transmit power of the BS is sufficiently large and the data size of the sampling message is quite small, thus the delay of broadcasting the sampling message can be ignored [12].

D. Energy Consumption Model

In our model, the energy consumption of each IoT device consists of status sampling energy consumption and status update energy consumption, which is given by

$$e_m(a_{m,t}) = a_{m,t} C_S + l_{m,t}(a_{m,t}) P_T, \quad (12)$$

where $a_{m,t} C_S$ is the energy consumption for status sampling with C_S being the sampling cost for generating the status packet and $l_{m,t}(a_{m,t}) P_T$ is the energy consumption for updating the status information.. Moveover, since the BS can have continuous power supply, we do not consider the energy consumption of the BS in our model.

E. Problem Formulation

Having defined the system model, next, we formulate an optimization problem whose goal is to minimize weighted sum of the AoI and energy consumption of all devices. The variable in this optimization problem for the BS is the sampling action indicator $\boldsymbol{a}_t$. The optimization problem is given by

$$\min_{\boldsymbol{a}_t} \sum_{t=1}^{T} \sum_{m=1}^{M} \left(\gamma_{\mathrm{A}} \Phi_{m,t}(a_{m,t}) + \gamma_{\mathrm{E}} e_m(a_{m,t})\right) \quad (13)$$

$$\text{s. t.} \quad a_{m,t} \in \{0, 1\}, \forall m \in \mathcal{M}, \forall i \in \mathcal{I}, \quad (13a)$$

$$\sum_{m \in \mathcal{M}} a_{m,t} \leqslant I, \ \forall i \in \mathcal{I}, \quad (13b)$$

where $\boldsymbol{a}_t = [a_{1,t}, \ldots, a_{M,t}]$. γ_{A} and γ_{E} are weighting parameters that combine the value of AoI and energy consumption into an integrated cost function. (13a) guarantees that each device can only occupy at most one RB for status update. (13b) ensures that each uplink RB can be allocated to at most one device. The problem in (13) is challenging to solve by conventional optimization algorithms due to the following reasons. First, as the physical process monitored by each IoT device varies, the BS must dynamically determine a subset of IoT devices for status sampling and update. Since the changes in the physical process $\boldsymbol{x}_{m,t}$ are correlated in time, the BS must consider this temporal correlation for the optimization of the sampling policies $\boldsymbol{a}_t$ at different slots. However, traditional optimization methods, such as dynamic programming, can only deal with the temporal correlation without the time-varying variables [13]. To overcome this limitation, we propose an RUQL algorithm to analyze the temporal features of physical process so as to optimize the sampling policy $\boldsymbol{a}_t$ and minimize the sum of AoI and energy consumption of all devices. Different from traditional learning algorithms, such as Q-learning with a fixed learning rate, the proposed method can dynamically adjust the learning rate. For instance, as the environment changes, the BS can increase the learning rate to obtain the dynamics of the varying environment, thereby improving the performance for optimization of the sampling policy.

III. Repeated Update Q-learning Method for Optimization of Sampling Policy

In this section, we introduce the proposed approach to solve problem (13). We first introduce the components of the learning algorithm. Then, we explain how to use it to solve the problem.

A. Components of Repeated Update Q-learning Method

The proposed RUQL algorithm consists of four basic components: a) agent, b) state, c) action, and d) reward function. Let $\mathcal{S}$ be the discrete set of environment states and $\mathcal{A}$ be the discrete set of actions available to the BS at each time slot. The components of the proposed algorithm are specified as follows:

- **Agent**: Our agent is the BS that selects a subset of devices to sample the physical process and collects the sampled packet so as to minimize the weighted sum of the AoI and energy consumption of all devices.
- **State**: Each environment state $\boldsymbol{s}_t \in \mathcal{S}$ at time slot t represents the AoI at each device $\boldsymbol{s}_t = [\phi_{1,t}, \ldots, \phi_{M,t}]$. In this time-slotted model with unit slot length τ, the value of $\phi_{m,t}$ is a positive integer that satisfies $\phi_{m,t} \in [0, \tau, 2\tau, \ldots, \phi_{\max}]$ and hence, the environment states that are discrete and finite can be recorded in Q-table.
- **Action**: The actions of the BS represent the subset of devices selected to sample the physical process. An action of the BS at time slot t can be defined as $\boldsymbol{a}_t = [a_{1,t}, \ldots, a_{M,t}]$ with $\boldsymbol{a}_t \in \mathcal{A}$ satisfying $\sum_{m=1}^{M} a_{m,t} \leqslant I$.
- **Reward**: Based on state $\boldsymbol{s}_t$ and chosen action $\boldsymbol{a}_t$, the reward function is given by
$$R(\boldsymbol{s}_t, \boldsymbol{a}_t) = -\left(\sum_{m=1}^{M}(\gamma_{\mathrm{A}}\Phi_{m,t}(a_{m,t}) + \gamma_{\mathrm{E}} e_m(a_{m,t}))\right), \quad (14)$$
where $\sum_{m=1}^{M}(\gamma_{\mathrm{A}}\Phi_{m,t}(a_{m,t}) + \gamma_{\mathrm{E}} e_m(a_{m,t}))$ is the objective function. Note that, $R(\boldsymbol{s}_t, \boldsymbol{a}_t)$ increases as the weighted sum of the AoI and energy consumption of all devices decreases, which implies that the agent will maximize the reward so as to minimize the weighted sum of the AoI and energy consumption.

B. Repeated Update Q-learning for Optimization of Sampling Policy

Given the components of the proposed RL algorithm, next, we introduce how to use the proposed algorithm to solve problem in (13). At each time slot, the BS chooses an action $\boldsymbol{a}_t$ from $\mathcal{A}$. After executing the selected action $\boldsymbol{a}_t$, the BS collects the sampled packet related to the dynamics of the physical process and then, observe the environment state $\boldsymbol{s}_t$. To record the current state and actions, we define a Q-table which is represented by $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$.

The proposed algorithm adopts the Boltzmann based action selection policy to prevent the BS only selecting the action with the highest initialized value rather than the optimal action. The Boltzmann exploration strategy $\pi(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is given by [14]
$$\pi(\boldsymbol{s}_t, \boldsymbol{a}_t) = \frac{e^{\frac{Q(\boldsymbol{s}_t, \boldsymbol{a}_t)}{\theta}}}{\sum_{\boldsymbol{a}'_t} e^{\frac{Q(\boldsymbol{s}_t, \boldsymbol{a}'_t)}{\theta}}}, \quad (15)$$
where parameter θ determines the probability of exploration.

With a fixed learning rate, the BS always learn the rewards from the actions with high probability to be chosen, and as a result, being stuck in the policy-bias problem in dynamic environments. To enable the BS to learn from all sampling policies, the update rule of the proposed RUQL method is given by
$$Q(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}) = \zeta Q(\boldsymbol{s}_t, \boldsymbol{a}_t) + (1-\zeta)(R(\boldsymbol{s}_t, \boldsymbol{a}_t) + \gamma \max_{\boldsymbol{a}'_t} Q(\boldsymbol{s}'_t, \boldsymbol{a}'_t)) \quad (16)$$
where $\boldsymbol{s}'_t$ and $\boldsymbol{a}'_t$ are the state and action after the BS performs action $\boldsymbol{a}_t$ under state $\boldsymbol{s}_t$. $\zeta = (1-\alpha)^{\frac{1}{\pi(\boldsymbol{s}_t, \boldsymbol{a}_t)}}$ is the adaptive learning rate that controls the speed of learning with $\alpha \in (0, 1)$ being a constant. $\gamma \in (0, 1)$ is the discount factor that controls the relative value of future rewards. To show how the learning rate ζ affects the update of the Q-table, we state the following theorem.

Theorem 1. The RUQL update rule can be approximated with an error of $O(\alpha^2)$ as an instance of QL that repeatedly updates the reward for $\lfloor \frac{1}{\pi(\boldsymbol{s}_t, \boldsymbol{a}_t)} \rfloor$ times with α being a constant denoting the learning rate of the underlying QL update equation.

Proof: See Appendix A. ∎

From Theorem 1, we can see that, the influence of the learning rate in RUQL can be regarded as a simple modification where the BS updates Q-table multiple times in traditional QL algorithm. In this way, the BS can balance the rates of the update among the actions with different probabilities to be chosen and hence, effectively learning from all feasible actions and updating all actions so as to avoid resulting in a policy-bias problem. Theorem 1 also shows that, as the BS updates the Q-table multiple times, $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$ will approach to the maximum reward with an error of $O(\alpha^2)$ and hence, the convergence of RUQL is guaranteed.

The training process of the proposed algorithm can be divided into two stages: offline training and online training. For offline training, the BS can use the historical dataset that consists of the dynamics of realistic physical process in different time slots to generate a trained Q-table, and hence, avoiding a poor performance at the beginning of online training. In particular, for each step, the BS chooses a random environment state $\boldsymbol{s}_t$ from its historical dataset, and then selects an action $\boldsymbol{a}_t$ based on (15) to determine the subset of devices for status sampling and update. After that, the BS can compute the sum of the AoI and energy consumption of devices so as to obtain the reward according to (14). Based on the state $\boldsymbol{s}_t$, the selected action $\boldsymbol{a}_t$, and the obtained reward $R(\boldsymbol{s}_t, \boldsymbol{a}_t)$, the BS can update its Q-table using (16). The BS will perform this training step repeatedly until convergence. Different from choosing an environment state from historical dataset for offline training, at online training process, the BS must observe the actual environment state at current time slot and then use the trained Q-table to choose an action. As the dynamics of physical process changes, the BS can update the Q-table dynamically and adjust the learning rate to obtain more information from the time-varying environment.

From the training process, we can see that, the delay and energy consumption for updating the Q-table can be negligible. This is because the BS that is equipped with powerful computational ability and continuous power supply only needs to calculate the reward once at each time slot. The training process also show that the solution obtained by the proposed method might not be the optimal solution. This is because, as the dynamics of the environment changes, the BS can only use

Algorithm 1 Repeated Update Q-learning method

Input: The environment state $\mathcal{S}$, the action space $\mathcal{A}$.
Output: The resource allocation strategy.
1: Initialize $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$.
2: Observe the current state $\boldsymbol{s}_t$.
3: **for** each time step **do**
4: Compute the policy $\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)$ using (15).
5: Choose an action $\boldsymbol{a}_t$ according to agent policy $\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)$.
6: Perform action $\boldsymbol{a}_t$ and observe reward $R(\boldsymbol{s}_t,\boldsymbol{a}_t)$ and next state s'.
7: Update $Q(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1})$ using (16).
8: Set $\boldsymbol{s}_{t+1}$=$\boldsymbol{s}'_{t+1}$.
9: **end for**

TABLE I
SIMULATION PARAMETERS [16]

Parameters	Values	Parameters	Values
M	11	τ	1 s
I	4	σ_N^2	-95 dBm
W	180 kHz	ξ_m	10 Hz
P_T	0.5 W	γ_{A}	0.001
C_S	0.3 mJ	γ_{E}	1
Z_m	10 bit	β	4

the trained Q-table to determine the sample policy in a new environment at each time slot. Without sufficient iterations in testing process, the BS cannot obtain the optimal solution. The proposed RUQL approach is shown in Algorithm 1.

IV. Simulation Results and Analysis

In our simulations, a circular network area having a radius $r = 100$ m is considered with $M = 11$ uniformly distributed IoT devices (unless stated otherwise) and one BS that is located centrally. Without loss of generality, the channel gain follows a Rayleigh distribution with unit variance. The values of other parameters are defined in Table I. Real data of the physical process that is sampled by each IoT device is obtained from the Center for Statistical Science at Peking University [15]. We compare our approach with the traditional QL method applied to the same system. All statistical results are averaged over 5000 independent runs.

Fig. 2 shows an example of the estimation of the physical process. In this figure, we can see that, as the index of the PM 2.5 changes, the proposed RUQL approach achieves better estimation accuracy compared to QL. This is due to the fact that the varying learning rate enables the BS to avoid policy-bias. Fig. 2 also shows that, as the physical process varies rapidly, the sampling frequency increases. This is because, as the index of the PM 2.5 changes rapidly, the estimation error of the proposed approach increases and hence, each device must increase the sampling frequency so as to collect more status information to capture the variation of the physical process. From Fig. 2, we can also see that, RUQL achieves an improvement in terms of the estimation accuracy compared to QL in non-stationary environments. This is because, as the environment dynamic changes over time, the proposed method enables the BS to update Q-table multiple times when executing an action under a new environment. Thus, the BS can learn more information from the dynamics of the environment and improve the estimation accuracy.

In Fig. 3, we show how the sum of AoI and the total energy consumption of all devices will vary as the number of IoT devices changes. From Fig. 3, we can see that, as the number of devices increases, the total energy consumption increases. This

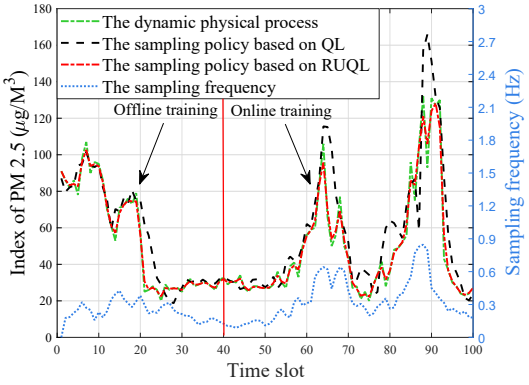

Fig. 2. The estimation of the dynamic physical process.

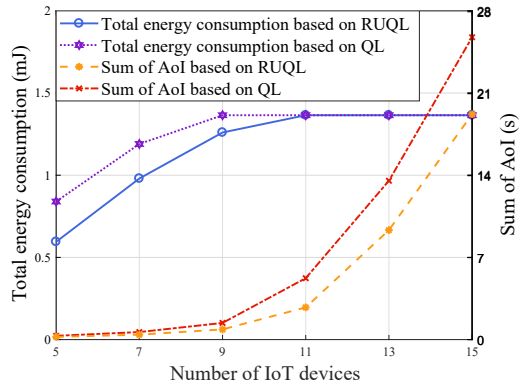

Fig. 3. The sum of AoI and total energy consumption as the number of IoT devices varies.

stems from the fact that, as the number of devices increases, the number of devices that must sample the physical process and transmit the sampled information to the BS increases, and, hence, the total energy consumption for status sampling and update increases. Fig. 3 also shows that, when the number of devices is larger than 11, the total energy consumption of all algorithms almost remains unchanged. This is due to the fact that the number of RBs is limited in our system. As the number of devices continues to increase, the number of devices that are selected by all algorithms for status sampling and update will be equal to the number of RBs, therefore, the total energy consumption of all algorithms almost remains unchanged. From Fig. 3, we can see that the proposed algorithm can reduce the total energy consumption by up to 11.5% compared to QL for the case with 4 RBs and 5 devices. This gain stems from the fact that the proposed algorithm enables the BS to adjust the learning rate to obtain more information from the executed actions as the dynamics change, thus capturing the variation of the physical process using less energy. Fig. 3 also shows that, as the number of devices increases, the sum of AoI increases. This is because the number of RBs is limited and, hence, with an increase of the number of devices, some devices may not be able to transmit the fresh updates to the BS immediately, which results in an increase of the sum of AoI. Moveover, as the number of devices continues to increase, the sum of AoI increases rapidly. This stems from the fact, as the number of devices is much larger than the number of RBs, most of the devices must wait until being allocated the RB so as to update the status which results in a great growth in terms of AoI. From

Fig. 3, we can see that the proposed algorithm can reduce the sum of AoI by up to 26.9% compared to QL for the case with 4 RBs and 15 devices. This gain stems from the fact that the proposed algorithm enables the BS to adjust the learning rate and avoid policy-bias to obtain a better performance.

V. Conclusion

In this paper, we have considered a real-time IoT system to capture the variation of a physical process. In the considered system, the physical process is modeled as a nonlinear dynamical equation which affects the sampling frequency of each device. Based on the proposed model, we have formulated an optimization problem that seeks to select a subset of IoT devices to sample the physical process so as to minimize the weighted sum of the AoI and total device energy consumption. To solve this problem, we have developed an RUQL algorithm that enables the BS to adjust the learning rate based on the chosen sampling policy to avoid the policy-bias problem, and hence, minimizing the objective in dynamic environments. Simulation results have shown that the proposed approach yields significant gains compared to conventional approaches.

Appendix

A. Proof of Theorem 1

To prove Theorem 1, we first show the update rule of traditional QL

$$Q_1(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})=Q_0(\boldsymbol{s}_t,\boldsymbol{a}_t)+\alpha(R(\boldsymbol{s}_t,\boldsymbol{a}_t)+\gamma\max_{\boldsymbol{a}'_t}Q_0(\boldsymbol{s}'_t,\boldsymbol{a}'_t)), \tag{17}$$

where α is learning rate. Then, we trace the update of $Q_{t+1}(\boldsymbol{s}_t,\boldsymbol{a}_t)$ for $\lfloor\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}\rfloor$ times. The first iteration result is given in (17). The second iteration result is

$$Q_2(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})=Q_1(\boldsymbol{s}_t,\boldsymbol{a}_t)+\alpha(R(\boldsymbol{s}_t,\boldsymbol{a}_t)+\gamma\max_{\boldsymbol{a}'_t}Q_1(\boldsymbol{s}'_t,\boldsymbol{a}'_t)).$$

Using the enumeration method, $\lfloor\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}\rfloor$ iteration result is

$$\begin{aligned}Q_{\lfloor\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}\rfloor}(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})=&(1-\alpha)^{\lfloor\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}\rfloor}Q_0(\boldsymbol{s}_t,\boldsymbol{a}_t)\\&+(1-\alpha)^{\lfloor\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}\rfloor-1}(R(\boldsymbol{s}_t,\boldsymbol{a}_t)+\gamma\max_{\boldsymbol{a}'_t}Q_0(\boldsymbol{s}'_t,\boldsymbol{a}'_t))\\&+...\\&+\alpha(R(\boldsymbol{s}_t,\boldsymbol{a}_t)+\gamma\max_{\boldsymbol{a}'_t}Q_{\lfloor\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}\rfloor-1}(\boldsymbol{s}'_t,\boldsymbol{a}'_t)).\end{aligned} \tag{18}$$

We consider the following two cases:

a) When the BS chooses the optimal action, i.e., $\boldsymbol{a}_t=\max_{\boldsymbol{a}'_t}Q(\boldsymbol{s}'_t,\boldsymbol{a}'_t)$, (18) can be rewritten as

$$\begin{aligned}&Q(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})=\\&(1-\alpha(1-\gamma))^{\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}}\left(Q(\boldsymbol{s}_t,\boldsymbol{a}_t)-\frac{R(\boldsymbol{s}_t,\boldsymbol{a}_t)}{1-\gamma}\right)+\frac{R(\boldsymbol{s}_t,\boldsymbol{a}_t)}{1-\gamma}.\end{aligned} \tag{19}$$

b) Otherwise, substituting $\zeta=(1-\alpha)^{\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}}$ into (18), we have

$$\begin{aligned}Q(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})=&(1-\alpha)^{\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}}Q(\boldsymbol{s}_t,\boldsymbol{a}_t)\\&+(1-(1-\alpha)^{\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}})(R(\boldsymbol{s}_t,\boldsymbol{a}_t)+\gamma\max_{\boldsymbol{a}'_t}Q(\boldsymbol{s}'_t,\boldsymbol{a}'_t)).\end{aligned} \tag{20}$$

Consider the Taylor series expansion of both equations at $\alpha=0$, and $(1-\alpha)^c=1-c\alpha+O(\alpha^2)$, (19) can be rewritten as

$$\begin{aligned}&Q(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})\\&=\left(1-\frac{\alpha}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}\right)Q(\boldsymbol{s}_t,\boldsymbol{a}_t)+\frac{\alpha(R(\boldsymbol{s}_t,\boldsymbol{a}_t)+\gamma Q(\boldsymbol{s}_t,\boldsymbol{a}_t))}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}+O(\alpha^2)\\&=\left(1-\frac{\alpha(1-\gamma)}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}\right)Q(\boldsymbol{s}_t,\boldsymbol{a}_t)+\frac{\alpha R(\boldsymbol{s}_t,\boldsymbol{a}_t)}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}+O(\alpha^2).\end{aligned} \tag{21}$$

Similarly, (20) can be rewritten as

$$\begin{aligned}&Q(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})\\&=\left(1-\frac{\alpha(1-\gamma)}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}\right)Q(\boldsymbol{s}_t,\boldsymbol{a}_t)+\frac{\alpha R(\boldsymbol{s}_t,\boldsymbol{a}_t)}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}+O(\alpha^2).\end{aligned} \tag{22}$$

From (21) and (22), we can see that, the RUQL update rule can be approximated with an error of $O(\alpha^2)$ as an instance of QL updating $\lfloor\frac{1}{\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)}\rfloor$ times. This completes the proof.

References

[1] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 4, pp. 3039–3071, Fourthquarter 2019.

[2] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the Internet of Things," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7468–7482, Mar. 2019.

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

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

[5] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Age of information and throughput in a shared access network with heterogeneous traffic," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018.

[6] X. Wu, J. Yang, and J. Wu, "Optimal status update for age of information minimization with an energy harvesting source," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 193–204, Mar. 2018.

[7] P. Zhang, Y. Yuan, H. Yang, and H. Liu, "Near-Nash equilibrium control strategy for discrete-time nonlinear systems with round-robin protocol," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 8, pp. 2478–2492, Aug. 2019.

[8] Y. Chen, S. Kar, and J. M. F. Moura, "Cyber-physical systems: Dynamic sensor attacks and strong observability," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, Apr. 2015.

[9] S. Karachontzitis, S. Timotheou, I. Krikidis, and K. Berberidis, "Security-aware max-min resource allocation in multiuser OFDMA downlink," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 529–542, Mar. 2015.

[10] Z. Wei, B. Li, and W. Guo, "Optimal sampling for dynamic complex networks with graph-bandlimited initialization," *IEEE Access*, vol. 7, pp. 150294–150305, Oct. 2019.

[11] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, to appear, 2020.

[12] L. Li, J. B. Song, and H. Li, "Dynamic state aware adaptive source coding for networked control in cyberphysical systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10000–10010, Aug. 2017.

[13] Y. Wang, M. Chen, Z. Yang, T. Luo, and W. Saad, "Deep learning for optimal deployment of uavs with visible light communications," *IEEE Transactions on Wireless Communications*, to appear, 2020.

[14] S. Wang, M. Chen, X. Liu, C. Yin, S. Cui, and H. V. Poor, "A machine learning approach for task and resource allocation in mobile edge computing based networks," *IEEE Internet of Things Journal*, to appear, 2020.

[15] B. Guo, B. Li, S. Zhang, and H. Huang, *Assessing Beijing's PM2.5 pollution: Severity, weather impact, APEC and winter heating*, Available Online: https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data/.

[16] M. Chen, M. Mozzaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE Journal on Selected Areas on Communications*, vol. 35, no. 5, pp. 1046–1061, May. 2017.