CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Journal Paper

Onboard Double Q-learning for Airborne Data Capture in Wireless Powered IoT Networks

Early Access

Kai Li*
Wei Ni
Eduardo Tovar*

*CISTER Research Centre

CISTER-TR-200402

2020/04/21

Onboard Double Q-learning for Airborne Data Capture in Wireless Powered IoT Networks

Kai Li*, Wei Ni, Eduardo Tovar*

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: kai@isep.ipp.pt, Wei.Ni@data61.csiro.au, emt@isep.ipp.pt

https://www.cister-labs.pt

Abstract

This letter studies the use of Unmanned Aerial Vehicles (UAVs) in Internet-of-Things (IoT) networks, where the UAV with microwave power transfer (MPT) capability is employed to hover over the area of interest, charging IoT nodes remotely and collecting their data. Scheduling MPT and data transmission is critical to reduce the data packet loss resulting from buffer overflows and channel fading. In practice, the prior knowledge of the battery level and data queue length of the IoT nodes is not available at the UAV. A new onboard double Q-learning scheduling algorithm is proposed to optimally select the IoT node to be interrogated for data collection and MPT along the flight trajectory of the UAV, thereby minimizing asymptotically the packet loss of the IoT networks. Simulations confirm the superiority of our algorithm to Q-learning based alternatives in terms of packet loss and learning efficiency/speed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LNET.2020.2989130, IEEE Networking Letters

Onboard Double Q-learning for Airborne Data Capture in Wireless Powered IoT Networks

Kai Li, *Senior Member, IEEE,* Wei Ni, *Senior Member, IEEE,* Bo Wei, and Eduardo Tovar

***Abstract*—This letter studies the use of Unmanned Aerial Vehicles (UAVs) in Internet-of-Things (IoT) networks, where the UAV with microwave power transfer (MPT) capability is employed to hover over the area of interest, charging IoT nodes remotely and collecting their data. Scheduling MPT and data transmission is critical to reduce the data packet loss resulting from buffer overflows and channel fading. In practice, the prior knowledge of the battery level and data queue length of the IoT nodes is not available at the UAV. A new onboard double Q-learning scheduling algorithm is proposed to optimally select the IoT node to be interrogated for data collection and MPT along the flight trajectory of the UAV, thereby minimizing asymptotically the packet loss of the IoT networks. Simulations confirm the superiority of our algorithm to Q-learning based alternatives in terms of packet loss and learning efficiency/speed.**

I. Introduction

Microwave Power Transfer (MPT)-enabled Internet of Things (IoT) networks provide a practical means to deploy wireless powered IoT nodes with no access to persistent power supplies [1]. However, IoT nodes far from an MPT transmitter can suffer low energy transfer efficiency due to the severe propagation loss of radio signals over long distance. Unmanned Aerial Vehicles (UAVs), also known as drones, equipped with MPT capability can provide an effective solution to this problem [2].

Each of the IoT nodes needs to be equipped with a wireless power harvester for harvesting radio frequency (RF) energy from the UAV to power its operations, as shown in Figure 1. The RF energy signals of the UAV can also carry data for the IoT node. The IoT nodes also sense their environments, generate data packets, and buffer the packets awaiting transmission. Consider a time-switching MPT technique [3] in this work. The MPT and data transmission can operate in the same radio frequency band but different time slots, so that each IoT node only needs a single RF chain to reduce the hardware cost.

This paper aims to holistically optimize the schedule of MPT and data transmission for a UAV to serve many IoT nodes deployed remotely with rechargeable batteries and without persistent power supply. The schedule is to select correct IoT nodes at correct times (to transfer energy to and collect data from) along the flight trajectory of the UAV, such that the packet loss of all the IoT nodes is minimized over a long period of time. Otherwise, an IoT node could be scheduled too early or too late, and the large distance between the IoT node and the UAV would reduce the energy transfer efficiency in the downlink and increase the packet loss due to failed transmissions in the uplink. Moreover, the IoT nodes with long data queues might not receive enough energy to upload their data in time, resulting in buffer overflow.

A practical setting is of particular interest, where an IoT node does not feed back its data queue and battery statuses until it is polled by the UAV to charge its battery and upload its data. In other words, the UAV does not have complete up-to-date knowledge of the entire network. A Markov Decision Process (MDP) has been typically formulated to optimize the collision-free flight trajectory of a UAV, target tracking, or power allocation, e.g., [4] and [5], typically under the assumption of the availability of the complete up-to-date network knowledge at the UAV. A resource allocation strategy was studied in [6] to reduce packet loss in energy harvesting powered sensor networks, where transmission probability and channel statistics were required before scheduling. Given the transition probabilities of the MDP, the resource allocations in [6] were solved by dynamic programming.

In [7], the scheduling of MPT and data transmission in a small-scale static wireless sensor network was formulated into an MDP. Q-learning was applied to derive the optimal solution to the MDP, when the data arrivals and channel conditions were unknown and so were the transition probabilities of the MDP. Unfortunately, with the growing number of IoT nodes, the knowledge of the UAV is increasingly differently outdated on different IoT nodes. This would destabilize the Q-learning process and even lead to the divergence of the learning. The reason is that a direct use of Q-learning in large-scale network settings can suffer from large packet loss approximation errors, as some IoT nodes may not be scheduled for a long time due to persistent poor channel conditions and subsequently low battery levels and the bias of using their local minimum value as the approximation of the optimum increases.

In this paper, we propose a new onboard scheduling algorithm using the concept of double Q-learning [8], which can effectively suppress the packet loss approximation errors, avoid overestimated rewards of selecting some IoT nodes and the biased estimation of IoT queue backlogs, and stabilize the learning process with convergence. Specifically, the first Q-learning process is designed to find an optimal scheduling strategy for MPT and data transmission

K. Li, and E. Tovar are with Real-Time and Embedded Computing Systems Research Centre (CISTER), 4249–015 Porto, Portugal (E-mail:{kai,emt}@isep.ipp.pt).

W. Ni is with Commonwealth Scientific and Industrial Research Organization (CSIRO), Sydney, Australia (E-mail: wei.ni@data61.csiro.au).

B. Wei is with Department of Computer and Information Sciences, Northumbria University, UK (E-mail: bo.wei@northumbria.ac.uk).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LNET.2020.2989130, IEEE Networking Letters

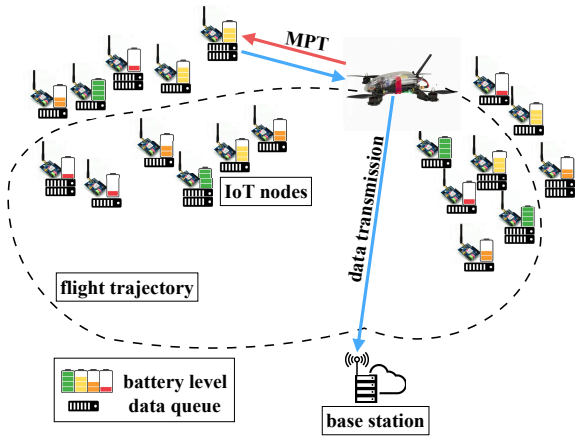

Fig. 1: UAV-assisted wireless powered IoT networks, where the UAV is employed to collect sensory data from the IoT nodes and recharge their batteries using MPT.

based on the (outdated) knowledge of the queue backlogs and battery levels of the IoT nodes. The second Q-learning is designed to generate a separate virtual scheduling process in which the unselected IoT nodes in the first Q-learning process can be selected based on replay memories. The two Q-learning processes are evaluated by each other for the next state. As a result, no IoT nodes are overlooked, and the packet loss approximation errors can be minimized even in a large-scale network with hundreds of IoT nodes.

II. Network Model

The UAV that acts as an airborne base station patrols along a pre-determined trajectory. The trajectory that consists of a number of waypoints is designed to cover N IoT nodes in the field [9]. The UAV applies MPT to remotely recharge the IoT nodes, where the received signal strength at the UAV is enhanced by receive beamforming to reduce bit error rate (BER). Although multi-user beamforming techniques, such as zero forcing and maximal ratio transmission, can be used to increase signal-to-noise ratio (SNR) of the communication channel, they are not considered in this work due to the requirement of real-time feedback on channel conditions.

Node i ($i \in [1, N]$) harvests energy from the UAV to charge its battery for powering its operations, e.g., sensing, computing and communication. The MPT efficiency depends on the distance between the UAV and the IoT node, and their antenna alignment at time t [7]. We assume the antenna of the IoT node is omni-directional. The beamforming is based on the locations of the IoT nodes. Coherent beamforming can be conducted to point a beam towards an IoT node. The UAV can acquire the accurate position of the node, typically by using its camera and image processing techniques, since line-of-sight is generally available between the UAV and the IoT node [10]. Therefore, the power transferred to node i is $P_i^{\mathrm{MPT}}(t) = \rho(d_i(t))P^{\mathrm{UAV}}$, where P^{UAV} is the constant transmit power of the UAV for MPT, and $\rho(d_i(t))$ is a charging efficiency factor that relies on the distance $d_i(t)$ between node i and the UAV at t. The battery readings of the IoT node, denoted by $e_i(t)$, are discretized into Ω levels, i.e., $e_i(t) \in [1, \Omega]$.

Let $\zeta(t)$ denote the waypoint of the UAV at t along the flight trajectory. The channel vector between the UAV and node i at t is $\mathbf{h}_i(\zeta(t))$, which can be obtained by channel reciprocity. Moreover, the modulation scheme of node i, denoted by $\phi_i(t)$, can be adapted for data transmission, where $\phi_i(t) \leq \Phi$, and Φ is the total number of modulations. The typical modulations are considered, such as BPSK, QPSK, and 8PSK, denoted by $\phi_i(t)$ = 1, 2, and 3, respectively, and $2^{\phi_i(t)}$ Quadrature Amplitude Modulation (QAM) with $\phi_i(t) \geq 4$.

The SNR between node i and the UAV using $\phi_i(t)$ is $\gamma_i(\phi_i(t))$ and $\gamma_i(\phi_i(t)) = \frac{\|\mathbf{h}_i(\zeta(t))\|^2 P_i(t)}{\sigma_0^2}$, where $P_i(t)$ is the transmit power of node i, and σ_0^2 is the noise power at the UAV. The transmit power of the IoT node depends on $\phi_i(t)$ and $\mathbf{h}_i(\zeta(t))$, and is given by $P_i(t) \approx \frac{\kappa_2^{-1} \ln \frac{\kappa_1}{\varepsilon}}{\|\mathbf{h}_i(\zeta(t))\|^2}(2^{\phi_i(t)} - 1)$, where κ_1 and κ_2 are channel constants [11].

III. Markov Decision Process Formulation and Onboard Double Q-learning Solution

A. MDP formulation

The resource allocation problem of interest is formulated as a discrete-time MDP with the network states consisting of the battery level $e_i(t)$ and queue length $q_i(t)$ of each IoT node, and the channel condition $\mathbf{h}_i(\zeta(t))$. The UAV takes actions to select the IoT nodes for energy harvesting and data transmission. The future battery energy and data queue length of every IoT node can be affected by the scheduling decisions of the UAV, which also leads to a non-negligible impact on the future actions of the UAV. Given random and independent data arrival and queueing process of sensory data at every IoT node, scheduling actions of the UAV can be modeled as a discrete-time random process in spite of partially controllable at the UAV.

The actions of the UAV can be optimized in a long-term stochastic control process, where the packet loss stemming from both overflowing buffers and unsuccessful data transmissions of the IoT nodes is minimized. Moreover, the optimization of the scheduling actions needs to be conducted in real-time over the MDP. The correlation between the scheduling decisions at different time slots needs to be captured, and to validate the long-term optimality of the resource allocation.

Each action of the UAV depends on the current network state $\mathcal{S}_\alpha$ which collects the battery levels $e_i(t)$ and queue lengths $q_i(t)$ of the IoT nodes ($i \in [1, N]$), and the channel conditions $\mathbf{h}_i(\zeta(t))$ between the IoT node i and the UAV. Given the H channel states and the maximum queue length of the IoT nodes D, the total number of MDP states is $[(\Omega+1)(D+1)H]^N$. The action $\mathcal{A}$ to be taken is to schedule one node to harvest energy via MPT and transmit data to the UAV at t, while specifying the modulation of the IoT node, i.e., $\mathcal{A} \in \Big\{(i, \phi_i(t)) : i = 1, 2, ..., N; \phi_i(t) \in \{1, 2, ..., \Phi\}\Big\}$.

Algorithm 1 Onboard Double Q-learning (ODQ) algorithm for UAV-assisted MPT and data collection

Initialize $\mathcal{S}_\alpha \in \mathcal{S}$, $k \in \mathcal{A}$, ϱ, and w
for time t_{learning} **do**
  while $\mathcal{S}_\alpha \in \mathcal{S}$ **do**
    The UAV selects the IoT node $k \in \mathcal{A}$.
    The UAV randomly updates either Q_A or Q_B.
    if The UAV $\rightarrow Q_A$ **then**
      IoT node $k^* \leftarrow \arg\min_k Q_A\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$.
      Based on (1), the UAV obtains $Q_A\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$.
    else
      IoT node $k^* \leftarrow \arg\min_k Q_B\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$.
      According to (1), the UAV obtains $Q_B\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$.
    end if
    The UAV calculates $\rightarrow C\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$.
    The UAV creates a record of $\mathcal{S}_\alpha$, $\mathcal{S}_\beta$, $Q_A^*\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\}$ and $Q_B^*\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\}$ in the action-value table.
    $\mathcal{S}_\alpha \leftarrow \mathcal{S}_\beta$ for the schedule in the next state.
  end while
end for

B. Onboard Double Q-learning Algorithm

An action-value function, denoted by $Q\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$, can be defined for minimizing the expected accumulated discounted cost when the UAV takes an action k following one of the resource allocation strategies thereafter, which transits state $\mathcal{S}_\alpha$ to $\mathcal{S}_\beta$. Specifically, the action-value function of each resource allocation strategy can be expressed in the form of the expected packet loss of the IoT nodes at the current state $\mathcal{S}_\alpha$ and the minimum of $Q\left\{\mathcal{S}_{\beta'} \middle| \mathcal{S}_\beta, k'\right\}$ over all future network states, which is

$$Q^*\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\} = (1-\varrho)Q^*\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\} + \varrho\left[C\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\} + \delta \min_{k' \in \mathcal{A}} Q\left\{\mathcal{S}_{\beta'} \middle| \mathcal{S}_\beta, k'\right\}\right] \quad (1)$$

where k' is the following action of k, $\mathcal{S}_{\beta'}$ is the future state of $\mathcal{S}_\beta$, and $\varrho \in (0, 1]$ is the learning rate. $C\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$ denotes the data packet loss resulting from buffer overflow and channel failed transmission when action k is taken to transit state $\mathcal{S}_\alpha$ to $\mathcal{S}_\beta$. The convergence of $Q\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$ to the optimal action-value function in (1) depends on ϱ.

It is known that Q-learning may overestimate actions that have been tried often. To address this issue, Algorithm 1 proposes the new Onboard Double Q-learning (ODQ) algorithm for UAV-assisted IoT networks, where the UAV calculates two action-value functions onboard, i.e., $Q_A\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$ and $Q_B\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$. When the UAV takes the action of scheduling an IoT node for MPT and data collection, one of the action-value functions in ODQ, e.g., $Q_A\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$, is updated with a value from the other $Q_B^*\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\}$ for the next state. Since $Q_B^*\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\}$ is updated throughout the scheduling optimization problem but with a different set of experience samples, it can be treated as an unbiased estimate for the value of this action. In addition, the UAV randomly assigns scheduling experiences (the states and actions that have been learned) to update one of the two action-value functions in ODQ.

Furthermore, the ODQ algorithm solves (1), even if the transition probability $\Pr\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$ is unknown. Given the learning time t_{learning}, the UAV observes the next state $\mathcal{S}_\beta \in \mathcal{S}$ based on the learning outcome of $\Pr\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$, which estimates the battery levels and queue lengths of all the IoT nodes in the network. Then, the UAV selects one IoT node to transfer energy and feed back data, namely, performing an action $k \in \mathcal{A}$. Accordingly, the immediate data packet loss due to buffer overflow or channel fading, i.e., $C\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$, and the next state $\mathcal{S}_\beta$ can be onboard calculated. Then, the UAV can update the value of $Q_A\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$ or $Q_B\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k\right\}$.

By applying (1), the UAV is able to schedule the optimal IoT node k^* for MPT and data collection at $\mathcal{S}_\alpha$. Given k^*, the next state $\mathcal{S}_\beta$ that is determined by the battery levels and data queue lengths of all the other non-scheduled IoT nodes can be accordingly updated. Moreover, ODQ creates a scheduling table on the UAV to store the results of double Q-learning, which records each state-action pair, i.e., $\mathcal{S}_\alpha$, $\mathcal{S}_\beta$, $Q_A^*\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\}$, and $Q_B^*\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\}$. Iteratively, ODQ explores all the states during t_{learning}, and determines the optimal IoT node at each state. The complexity of Algorithm 1 depends on the update of the action-value function, i.e., Q_A or Q_B in (1). The learning rate $\varrho \in (0, 1]$ ensures that the updates of the action-value function average over possible randomness in the packet loss and transitions in order to converge in the limit to the optimal action-value function. The convergence of ODQ that carries out double estimator is dependent on ϱ [12]. Each time Q_A or Q_B is updated, the UAV selects one of the N IoT nodes for MPT and data collection. Therefore, ODQ with low complexity is feasible to the online scheduling of UAV-assisted MPT and IoT transmission.

IV. Numerical Results

The data transmission is given as a sequence of time slots, where each IoT node can generate one data packet per time slot and put into the queue. Consider that the ground IoT nodes monitor urban environment [13], and each data packet can have a payload of 32 bytes. Nakagami-m fading channel model is considered. The BER of the channel has to be higher than 0.05% in order to decode the packet, where ε = 0.05%. However, the value of ε can be configured depending on the traffic type, Quality-of-Service (QoS) requirement, and the transmission capability of the UAV-assisted IoT network.

The UAV has a constant transmit power of 40 dBm for MPT, while the charging distance is 5.77m [2]. The UAV moves along a predetermined circular flight trajectory with radius of 500 meters, as considered in [11]. It is important

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LNET.2020.2989130, IEEE Networking Letters

TABLE I: The learning error of ODQ and Q-learning in terms of packet loss rate

Number of nodes	ODQ	Q-learning	Optimum	learning errors of ODQ	learning errors of Q-learning
10	0.017	0.017	0.012	0.005	0.005
15	0.023	0.023	0.016	0.007	0.007
20	0.02	0.029	0.018	0.002	0.011
25	0.028	0.035	0.022	0.006	0.013
30	0.037	0.043	0.026	0.011	0.017
35	0.053	0.062	0.043	0.01	0.019
40	0.066	0.091	0.05	0.016	0.041

to note that the proposed ODQ algorithm is generic, and can schedule MPT and data collection given any other flight trajectory of the UAV. The flight trajectory of the UAV needs to be meticulously designed so that all the IoT nodes can be accessible by the UAV.

We first consider a small-scale problem, where $N \in [10, 40]$, $\Omega = 5$, and $D = 6$, in order to show the superiority of the proposed ODQ algorithm over the Q-learning approach that is used in [7]. Moreover, a benchmark policy is achieved by dynamic programming [6], where the transition probability and the cost of all states are assumed to be known to the UAV in prior. Table I shows the packet loss rate and learning error of ODQ and Q-learning, where the learning error defines the difference between the optimal packet loss rate achieved by ODQ or Q-learning and dynamic programming. Here, the network packet loss rate is the amount of packet loss due to the data queue overflow and the poor channel condition. As observed, the average learning error of ODQ is 0.81%, which is lower than 1.61% of Q-learning when $N \in [10, 40]$. With the growth of state and action spaces, the learning error of ODQ increases by 3.2 times, while the learning error of Q-learning increases by 8.2 times.

For performance comparison, two non-learning scheduling policies are simulated. The first algorithm, referred to as "Longest Queue Scheduling Algorithm (LQSA)", greedily gives the highest priority of MPT and data transmission to the node with the most data in the queue. The second algorithm is referred to "Longest Queue Lowest Battery algorithm (LQLB)", where scheduling is based on the data queue length and the battery level of the node. LQLB gives the highest priority to the IoT node with either the longest queue or the lowest battery level.

We consider that the IoT nodes ($N \in [10, 150]$) are deployed in the area of a circle with a radius of one kilometer. The node has the maximum discretized battery level $\Omega = 20$ and queue length $D = 10$. Figure 2 presents the network packet loss rate with an increasing number of IoT nodes, where the data queue length is set to 10 packets. Specifically, the proposed ODQ scheme outperforms the

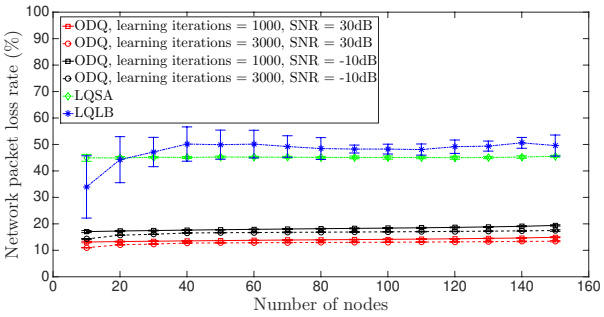

Fig. 2: Network packet loss rate with regards to the IoT network size, where the error bars are derived over 15 runs.

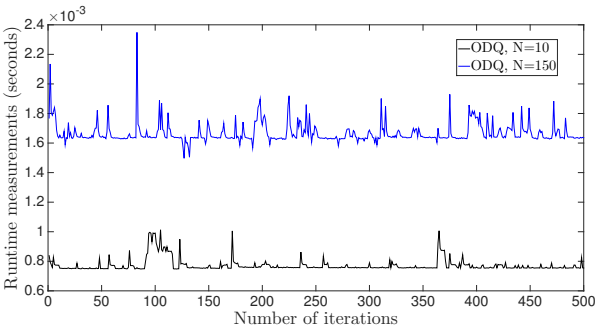

Fig. 3: Runtime measurements of ODQ.

two non-learning scheduling policies. The reason is that ODQ learns the IoT nodes' data queue, battery, and channel state to minimize the packet loss of the entire network. Furthermore, ODQ with more learning iterations or higher average channel gain achieves lower packet loss. This is reasonable, since a large t_{learning} allows $Q^*\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\}$ in ODQ to be derived with more states and actions, while a high $\mathbf{h}_i(\zeta(t))$ leads to a small $Q^*\left\{\mathcal{S}_\beta \middle| \mathcal{S}_\alpha, k^*\right\}$.

Figure 3 shows the runtime of ODQ, where the number of IoT nodes is 10 or 150. The runtime of ODQ is around 0.8 ms when $N = 10$, and increases to 1.7 ms when $N = 150$. This is because the increased number of IoT nodes enlarges the state space. Nevertheless, the relative increase in the runtime is much slower than that in the number of nodes. This is because the action space does not grow dramatically with the number of nodes. In particular, given a network state, the actions that the UAV can take are limited by its current position and the number of IoT nodes within the radio coverage.

Figure 4 studies the network packet loss rate with regards to data queue length of the IoT node, where the number of IoT nodes is set to 100. In particular, when $D = 20$, ODQ achieves 29.5% and 28.2% lower packet loss rates than LQSA and LQLB, respectively. This is because LQSA and LQLB schedule the IoT nodes based on either the longest data queue or the lowest battery level. The scheduled node can suffer from low energy harvesting efficiency and high packet loss rate, due to the severe propagation loss of MPT and wireless transmission over long distance. In contrast, ODQ schedules the IoT nodes to minimize the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LNET.2020.2989130, IEEE Networking Letters

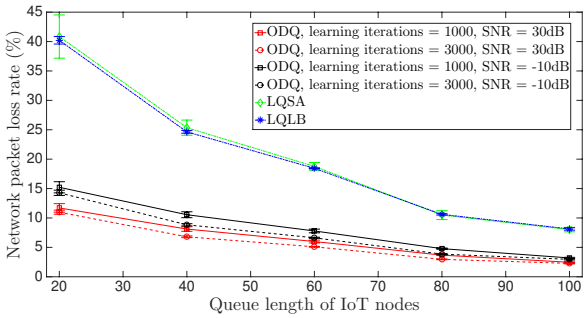

Fig. 4: Network packet loss rate with regards to data queue length of the IoT node, where the error bars are derived over 15 runs.

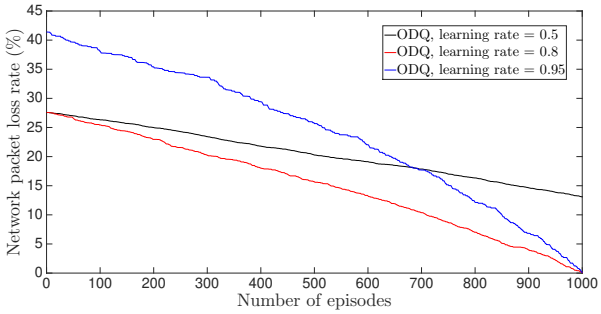

Fig. 5: Network packet loss of ODQ with regards to the episodes.

data packet loss of the network, where the UAV learns the state transitions of the network and takes the optimal actions at every moment. Moreover, given $t_{\text{learning}} = 1000$ and $\mathbf{h}_i(\zeta(t)) = -10$dB, from $D = 20$ to $D = 100$, the network packet loss rate of ODQ drops by 11.8%, This confirms that ODQ significantly reduces data queue overflow for all the IoT nodes when the data queue length increases.

In addition, the learning rate in double Q-learning can affect the convergence time of ODQ since the action-value function is approximated with asymptotic convergence. Figure 5 plots the network packet loss rate of ODQ from episode 0 to episode 1000 given the learning rate ϱ = 0.5, 0.8, or 0.95. At the beginning of the learning, the network packet loss rate of ODQ is about 27.3%. Due to double Q-learning, the packet loss of ODQ significantly drops with the increase of the episodes. Particularly, the performance of ODQ with $\varrho = 0.8$ falls to 0 at episode 1000. However, ODQ with $\varrho = 0.5$ requires more learning iterations to converge.

Reinforcement learning is known to have a tradeoff between the learning rate and the approximation accuracy of the action-value functions [12]. As shown in Figure 5, the high learning rate of ODQ, i.e., ϱ = 0.95, can lead to fast convergence, however at a cost of the approximation accuracy of the action-value functions. The estimated packet loss rate is higher than the one with small ϱ. Therefore, the learning rate of ODQ needs to be carefully configured to fulfil performance requirement of the UAV-assisted wireless powered IoT network.

V. Conclusion

This letter studies scheduling MPT and data transmission for preventing buffer overflow and battery depletion of the ground nodes in UAV-assisted wireless powered IoT networks. We propose ODQ for the UAV to optimally decide the IoT node for data collection and MPT along the flight trajectory of the UAV, thereby minimizing asymptotically the data packet loss. ODQ exploits double Q-learning on the UAV to significantly reduce the approximation errors resulting from overestimation of the scheduling decision.

Acknowledgements

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDB/04234/2020); also by the Operational Competitiveness Programme and Internationalization (COMPETE 2020) under the PT2020 Partnership Agreement, through the European Regional Development Fund (ERDF), and by national funds through the FCT, within project(s) POCI-01-0145-FEDER-029074 (ARNET).

References

[1] K. W. Choi, A. A. Aziz, D. Setiawan, N. M. Tran, L. Ginting, and D. I. Kim, "Distributed wireless power transfer system for internet of things devices," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2657–2671, Aug. 2018.

[2] J. Xu, Y. Zeng, and R. Zhang, "UAV-enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5092–5106, Aug. 2018.

[3] C. Qin, W. Ni, H. Tian, R. P. Liu, and Y. J. Guo, "Joint beamforming and user selection in multiuser collaborative MIMO SWIPT systems with nonnegligible circuit energy consumption," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 3909–3923, May 2018.

[4] K. Li, W. Ni, E. Tovar, and A. Jamalipour, "On-board deep Q-network for UAV-assisted online power transfer and data collection," *IEEE Trans. Veh. Technol.*, Oct. 2019.

[5] L. Xiao, C. Xie, M. Min, and W. Zhuang, "User-centric view of unmanned aerial vehicle transmission against smart attacks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3420–3430, Apr. 2018.

[6] K. Li, W. Ni, L. Duan, M. Abolhasan, and J. Niu, "Wireless power transfer and data collection in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2686–2697, Mar. 2018.

[7] K. Li, W. Ni, M. Abolhasan, and E. Tovar, "Reinforcement learning for scheduling wireless powered sensor communications," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 2, pp. 264–274, Jun. 2018.

[8] H. Van Hasselt, "Double Q-learning," in *Advances in Neural Information Processing Systems*, 2010, pp. 2613–2621.

[9] Z. M. Fadlullah, D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, "A dynamic trajectory control algorithm for improving the communication throughput and delay in UAV-aided networks," *IEEE Netw.*, vol. 30, no. 1, pp. 100–105, Jan. 2016.

[10] L. Li, Y. Xu, Z. Zhang, J. Yin, W. Chen, and Z. Han, "A prediction-based charging policy and interference mitigation approach in the wireless powered internet of things," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 2, pp. 439–451, Feb. 2019.

[11] K. Li, W. Ni, X. Wang, R. P. Liu, S. S. Kanhere, and S. Jha, "Energy-efficient cooperative relaying for unmanned aerial vehicles," *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1377–1386, Jun. 2016.

[12] E. Even-Dar and Y. Mansour, "Learning rates for Q-learning," *Journal of machine learning Research*, vol. 5, no. Dec, pp. 1–25, 2003.

[13] K. Li, C. Yuen, B. Kusy, R. Jurdak, A. Ignjatovic, S. S. Kanhere, and S. Jha, "Fair scheduling for data collection in mobile sensor networks with energy harvesting," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1274–1287, 2018.