Optimized Scheduling of Packets with Deep Sensing Irregular Slotted Aloha Influenced by Channel Knowledge for Wireless-Powered IoT Networks

1st Muhammad Tauseef Mushtaq
Department of Electrical and Information Engineering
Politecnico di Bari
Bari, Italy
m.mushtaq@phd.poliba.it

2nd Nicola Cordeschi
Department of Electrical and Information Engineering
Politecnico di Bari
Bari, Italy
nicola.cordeschi@poliba.it

3rd Alfredo Grieco
Department of Electrical and Information Engineering
Politecnico di Bari
Bari, Italy
a.grieco@poliba.it

4th Gennaro Boggia
Department of Electrical and Information Engineering
Politecnico di Bari
Bari, Italy
gennaro.boggia@poliba.it

***Abstract*—Wireless Powered Communication Networks (WPCNs) play a crucial role in critical operations and disaster management due to their ability to operate efficiently with minimal energy requirements. Additionally, in WPCN the sensors can harvest energy from a hybrid access point (HAP). However, designing efficient and robust scheduling for WPCNs is challenging due to limited energy sources and constrained nature of devices. Due to sporadic and generally event-driven nature of traffic of such networks and limited-size packets, random algorithms such as repetitive ALOHA or it's upgraded versions such as Deep Sensing Irregular Repetitive Slotted ALOHA (DS-IRSA) are particularly effective in WPCN. In this paper, our goals are to design a harvest or access protocol that uses mini-slots to sense the channel and control the number of repeated packets desired to be sent by each sensor in each frame. Specifically, differently from DS-IRSA, that uses mini-slots for traditional carrier sense before allowing actual data transmission, we utilize these minis-lots not only at the MAC layer to sense the number of packets but also, at the physical layer to sense the channel between the sensor and HAP and leverage this information to boost performance. Channel information plays a vital role in identifying sensors with unfavourable channel conditions, enabling the algorithm to make optimized decisions. Building upon the advantages of DS-IRSA, we propose a deep reinforcement learning algorithm that determines the optimal number of replicas for each sensor, as well as the optimized number of slots for energy harvesting.**

***Index Terms*—DS-IRSA, WPCN, HAP, IoT Sensors, Energy Harvesting**

I. Introduction

WPCN are the networks that play a critical role in areas hit by disaster. The importance of WPCNs lies in their ability to minimize logistical resource requirements and harness energy from various sources. Literature suggests that designing a perfect scheduling algorithm for packet transmission is almost impossible. Therefore, there is an opportunity for improvement in scheduling packets in a more robust and efficient manner. With hundreds of sensors ready to connect and send data to the access point, it is important to schedule their packets while simultaneously performing energy transmission. A resource allocation solution such as available for mobile communication is more time and resource-consuming due to sporadic nature of these systems. DS-IRSA [3] uses mini-slots to sense the environment before allowing actual data transmission. This paper aims to design a harvest or access protocol that uses mini-slots to sense the channel and number of repeated packets each sensor desires to send over each frame. These mini-slots are bursts of energy called jamming signals or beacon signals.

We provide a brief description of the literature that is our focus. Starting with two energy harvesting papers [1], [2], papers focus on simultaneously scheduling the data packets and energy harvesting. In this context, [2] focuses on the probabilistic analysis of available slots in a frame. The main objective of this paper is to utilize the Idle slots for harvesting along with reducing the probability of collision. On the other hand, [1], deploys a Q-learning approach to find the best possible scheduling of all users. Similarly, another method is to introduce the mini-slots before the transmission [3]. Mini-slots are small jamming signals from the users to inform HAP and other users about the slots where they will send replicas. In this way, users and HAP have prior knowledge of slots availability, which enables them to plan their replicas accordingly. Exploiting the mini-slot concept, [3] uses the DRL to learn number and pattern of replicas of users for better throughput. It adopts the Markov properties of the system and uses an optimization policy called Proximal Policy Optimization (PPO). Implementing PPO helps the HAP to solve an optimization problem that provides the base for decoding the packets.

In this paper, we consider an energy harvesting scenario with sensors transmitting their packets to a HAP which is capable of simultaneously decoding the packets and transmitting energy to the sensors. Our concept involves utilizing these mini-slots at the network layer to not only sense the number of packets but also perform channel sensing (between a sensor and HAP) at the physical layer. Channel information is of key importance here. It can help the algorithm to recognize the sensors that do not enjoy favorable channels with HAP. Using the qualities of DS-IRSA, our deep reinforcement learning algorithm achieves its two objectives; one deciding the number of replicas for each sensor and finding the optimal slots for energy harvesting. In this way, we achieve our objectives of lower the computational cost of the sensors and being more aware of the environment . That eventually leads to better decisions.

The rest of the paper is as follow. Section II describes the system model of wireless powered IoT network. Section III and IV present the two different algorithms which are able to manage the number of replicas for better throughput. In section V, we discuss the results obtained from these algorithms while section VI concludes the work and provides future suggestions.

II. System Model

We consider set of sensors deployed in the field that are named as $u_1, u_2, ..., u_N$. These sensors are always in saturated mode i.e. they always have data to sent. The access point has the dual role of not only receiving packets from sensors but also provide them power through energy harvesting. We call our access point HAP. It has to schedule not only receiving of the packets but also needs to consider available battery as well. Each sensor has battery unit denoted by β. In our case, each sensor needs one battery unit β to send one packet and can store maximum of three units. Hence users are limited to send three replicas at max in each frame. Each sensor has a channel between itself and HAP denoted by h_n. We categories our channel at MAC layer based on physical layer analysis ranging from worst to best.

Considering the above scenarios we start with the following objectives:

- Energy harvesting is an important ingredient of WPCN
- We need to optimize the system and find a balance between energy harvesting and scheduling the packet transmission

There are multiple methods available in the literature. We explore two of these methods in our system model.

A. Scheduling of Packets with JAL Influenced by Channel and Power for WPCN

Energy harvesting is an essential component of WPCN. However, with thousand of sensor attempting to compute with a HAP as in massive Machine Type Communication (mMTC), it becomes difficult to accommodate every sensors. For this purpose, we propose a Q-Learning algorithm which optimizes the the number of replicas in IRSA environment. The decision depends on three parameters; number of available slots, available energy of the sensor and channel quality. In this context, we design JAL based on the work represented in [1]. Section III describes this algorithm with more details.

B. Scheduling of Packets with DS-IRSA Influenced by Channel for WPCN

Key Points of DS-IRSA with channel information are as follows.

- With thousands of sensors ready to connect and send data to the access point, the important point is to schedule their packets along with energy transmission.
- With thousands of sensors ready to connect and send data to the access point, the crucial aspect is to schedule their packets while considering energy transmission.
- That is why a random and repetitive nature algorithm such as repetitive ALOHA or upgraded versions such as DS-IRSA uses mini-slots to sense the environment before allowing actual data transmission.

III. Joint Action Learner (JAL) with Power and Channel Constraints

Before providing the details of JAL, we provide a summary of the Q-Learning. Q-Learning takes the approach of reinforced learning and works with reward and action methods. We define state $S_t = \{s_1^t, s_2^t, ...s_M^t\}$ as the combination of power available and channel capacity at each user such that $s_1^t = (p_1^t, c_1^t)$. Where p_1^t is the power available at the sensor and c_1^t is the channel quality of the sensor. It can be noted that even though channel quality between HAP and sensor remains best, the sensor's capability of sending replicas can be limited by available power. Similarly, the energy harvesting capability of sensors can be a limiting factor in deciding the number of replicas. Keeping all this in the loop, our algorithm decides a joint action for the users who intend to send their data in a specific frame. The computational overhead of the algorithm is performed at HAP since it is not affected by the limited amount of energy as compared to sensors. The algorithm decides the number of replicas for each user and decides the specific actions for each user defined as $A_t = \{a_1^t, a_2^t, ...a_M^t\}$. Q-Learning updates and optimizes the Q-values using Bellman's equation defined as

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha(r(s_t, a_t) + \gamma \mathbf{max} Q(s_{t+1}, a)). \quad (1)$$

where $r(s_t, a_t)$ is the reward collected after taking a specific action.

JAL is a Q-Learning-based algorithm that uses the data provided by the sensors to decide the number of replicas such that the throughput is maximized. HAP is responsible for running the JAL, which leads to a lower computational cost of running a learning algorithm for sensors whose power is scarce. After learning through JAL, HAP provides the guidelines to the sensors for the next transmission. A summary of our JAL algorithm is provided below in the Algo. 1.

Algorithm 1 JAL with Power and Channel Constraints

Initialize Parameters α, γ
Initialize $Q(S, A)$ randomly
for `For each frame` t **do**
  HAP collects s_i^t
  Obtain the joint state $S_t = \{s_1^t, s_2^t, ...s_M^t\}$
  Generate random number x
  if $x < \epsilon$ **then**
    Select a joint action randomly
  else
    Select a joint action by solving
    $A_t(S_t) = argmax_{A \in \mathcal{A}} Q(S_t, A_t)$
  end if
end for
for device $i \in M$ **do**
  Randomly select data slots for action $a_i^t \in A_t$
  Collect the reward r_i^t
  Observe the next state S_i^{t+1}
end for
Calculate the joint reward: $R_t = \sum_{i=1}^{M} r_i^t$
Obtain the next joint state: $S_{t+1} = \{s_1^{t+1}, s_2^{t+1}, ..., s_M^{t+1}\}$
Find $max_{A \in \mathcal{A}} Q(S_{t+1}, A)$
Update the Q-value using Bellman's equation

IV. Deep Sensing IRSA with Channel Constraints

As mentioned in the literature related to DRL, we can observe that Q-Learning works fine as long as the system does not need re-scaling. This is not the case when we are dealing with WPCN since there is always a possibility of new sensors joining the network. With this limitation, JAL also inherits the issue of re-scaling. To fulfill this need, we propose DS-IRSA which is probabilistic in nature and do not need a tabular form of data. We not only take into account the fact that we need to accommodate energy harvesting slots but also the decision or learning of algorithm dependents on channel quality between sensor and HAP. These code-words are dealt with as pilot signals and hence it is convenient for the physical layer to sense the channel. Meanwhile, the MAC layer senses the number of replicas a sensor is intending to send. We implement DS-IRSA using the following environment.

A. The Actions

The set of action of users $m \in M$ is defined as A_m and it has two parts:

- The action sequence in sensing phase A_m^{sens} with L number of mini-slots. The vector contains the binary values 0 or 1 which represents the jamming signal. We read 1 when a jamming signal is send and 0 otherwise.
- A_m^{trans} represents the action during the transmission phase where users send their replicas at different slots in one frame.

B. The States

The state of the user represents observed values of different parameters, its components are provided below:

- The first component of the state is the observed code-word sent by the user during the sensing phase.
- The second component is the index of the current mini-slot
- The third component is the quality of the channel observed during the sensing phase.

C. The Reward

We donate the reward as R_m for each user. It represents the successful packet transmitted by that user during the frame. HAP calculates the reward at the end of each frame.

Summary of our algorithm is provided below in Algo. 2.

Algorithm 2 DS-IRSA with Mini-slots

Training and Sensing Phase
Initialize the number of Users and mini-slots u, m
Generate random set of possible code-word for Each User **S**
for `For each time step` t **do**
  HAP senses the jamming signal and collects the code-words matrix **S**
  Feed the received **S** to DRL NN
  Train the DRL using PPO Baseline3
end for
Decision Phase
for Users u **do**
  Assign new code-word to each user
  Sense the number of collisions with new code-word
  Repeat the code-word generation until optimized
end for

V. Results and Discussion

In this section, we first demonstrate the results obtained from JAL, whose base is Q-Learning. We see how varying the channel quality and available battery can affect the number of replicas. Then we proceed to our deep sensing-based IRSA algorithm that uses a training phase to learn the behavior of different users accompanying their channel quality as well. After the learning phase, DS-IRSA based on the training and available channel information directs the users to send their replicas such that throughput is maximized.

Fig. 1 provides the probability of the total number of replicas calculated by running 40,000 episodes in a Python script. In the current scenario, we consider 4 users and 4 slots, with random battery units available for each user. We grade the channel calculations and assign the channel of each user a value ranging from 0 to 4. The channel grade works in following manner: 0=bad, 1=medium, 2=good, 3= best. Similarly, we define our battery with unit β available for each user which ranges from $\beta = 1 - 3$. As a result, JAL acquires a scenario, and it can make decisions based on the available information regarding battery and channel. For example, if a user intends to send two replicas and it experiences the best channel with HAP, JAL can ask this user to send only one replica. Another user can utilize it who is not experiencing great channel quality and the risk of losing the packet is great. After making such decisions, Fig. 1 provides that on average 45% of the time, there will be a total of three number replicas in the scenario which has 4 users and 4 slots. Thus, applying successive interference cancellation is simple with the optimized and perfectly slotted replicas.

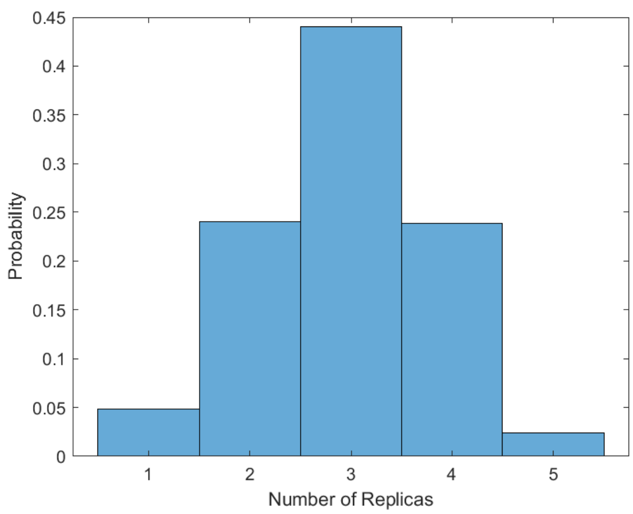

Fig. 1. Number of Replicas for Each Slot with JAL

For DS-IRSA, we aim to select code-words for two users which results in mazimum throughput. We run a Python script with 100,000 episodes whereas every episode consists of two phases that is sensing phase and transmission phase. During the sensing phase, the algorithm learns the channel quality and number of replicas each user intends to send from the jamming signal. Fig. 2 and 3 provide the selection of code word for 100,000 iterations after DS-IRSA has derived the code-word for each user. We can see that for $U1$ DS-IRSA selects 01 while it selects 10 for $U2$. There appears to be a conflict for some episodes where each user intends to send replicas at both slots. As the episodes progress and DS-IRSA learns better, we experience a better throughput. Hence we can observe an optimized code-word can enhance the throughput.

VI. Conclusion and Future Work

In this paper, we have addressed the challenge of slot selection in an energy-harvesting IoT system, presenting two distinct approaches. As a solution, we have introduced two different Deep Reinforcement Learning (DRL) algorithms. Our proposed DRL algorithms incorporate crucial parameters such as available battery capacity and channel quality, allowing the system to develop an awareness of the environment. For future work, we can achieve more realistic results by incorporating the energy harvesting models which use the well established channel and energy models.

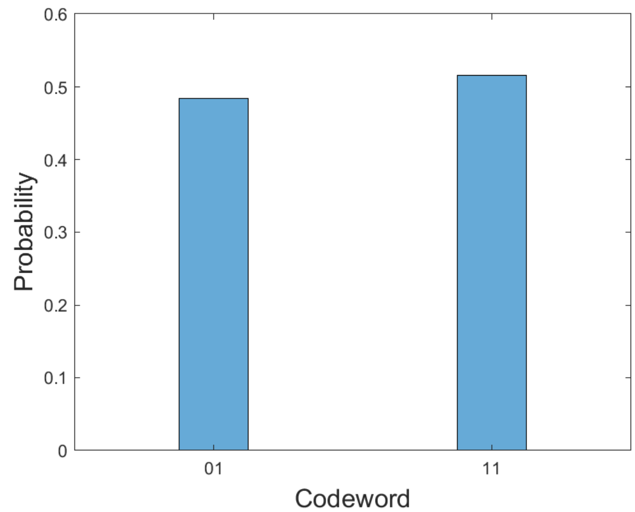

Fig. 2. Code-word Intention of User 1

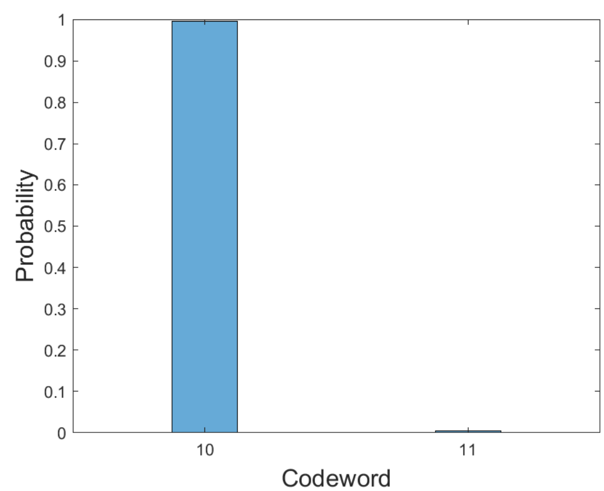

Fig. 3. Code-word Intention of User 2

References

[1] Y. Li, K. Chin, ”Energy-Aware Irregular Slotted Aloha Methods for Wireless-Powered IoT Networks”, IEEE INTERNET OF THINGS JOURNAL, VOL. 9, NO. 14, JULY 15, 2022.

[2] H. Choi, W. Shin, M. Levorato, H. Vincent Poor, ”Harvest-Or-Access: Slotted ALOHA for Wireless Powered Communication Networks”, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 68, NO. 11, NOVEMBER 2019.

[3] I. Hmedoush, C. Adjih, P. Mühlethaler, ”Deep Learning, Sensing-based IRSA (DS-IRSA): Learning a Sensing Protocol with Deep Reinforcement Learning”, 23 Sep 2022, https://hal.inria.fr/hal-03744126v2.

[4] N. Cordeschi, F. Rango, M. Tropea, ”Exploiting an optimal delay-collision tradeoff in csma-based high-dense wireless systems”,IEEE TRANSACTIONS ON NETWORKING, VOL. 29, NO. 5, JUNE 2021.

[5] F. Rango, N. Cordeschi, F. Ritacco, ”Applying Q-learning approach to CSMA Scheme to dynamically tune the contention probability”, IEEE 18th Annual Consumer Communications and Networking Conference (CCNC) March 2021, Las Vegas, NV, USA.