LoRaDRL: Deep Reinforcement Learning Based Adaptive PHY Layer Transmission Parameters Selection for LoRaWAN

Inaam Ilahi*, Muhammad Usama*, Muhammad Omer Farooq†, Muhammad Umar Janjua* and Junaid Qadir*
*Information Technology University, Lahore, Pakistan
†Department of Systems and Computer Engineering, Carleton University, Canada

2020 IEEE 45th Conference on Local Computer Networks (LCN) | 978-1-7281-7158-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/LCN48667.2020.9314772

***Abstract*— The performance of densely-deployed low-power wide-area networks (LPWANs) can significantly deteriorate due to packets collisions, and one of the main reasons for that is the rule-based PHY layer transmission parameters assignment algorithms. LoRaWAN is a leading LPWAN technology where LoRa serves as the physical layer. Here, we propose and evaluate a deep reinforcement learning (DRL)-based PHY layer transmission parameter assignment algorithm for LoRaWAN. Our algorithm ensures fewer collisions and better network performance compared to the existing state-of-the-art PHY layer transmission parameter assignment algorithms for LoRaWAN. Our algorithm outperforms the state of the art learning-based technique achieving up to 500% improvement of PDR in some cases.**

I. Introduction

Over the next few years, the Internet of things (IoT) networks are expected to grow exponentially. Most IoT end devices (EDs) are expected to be: (i) low-cost, (ii) long-range, and (iii) ultra-low-power. These EDs will provide vital information for intelligent decision making in many smart applications in diverse fields such as health-care systems, inventory management, and smart parking. The large number of EDs deployed in such networks poses significant challenges such as low packet delivery ratio (PDR) and high overall power consumption. Due to these challenges, there is a need for an effective communication technique that can enable simultaneous transmissions from several devices while (i) reducing packet collisions and (ii) keeping power consumption low.

Long-range (LoRa) is a low-power wide-area-network (LPWAN) protocol that enables multiple simultaneous transmissions through customization of PHY layer transmission parameters. LoRaWAN [1] is the open MAC layer specification for LoRa. LoRaWAN uses pure ALOHA as the MAC protocol primarily because a simple protocol better suits low-power EDs. As ALOHA does not sense a communication channel before transmission, therefore with an increase in data traffic load the network performance deteriorates. In the light of the above discussion, LoRa's ability of PHY layer parameter customization can become helpful as an intelligent PHY layer transmission parameter assignment algorithm can not only make up for ALOHA's performance issues, it can also result in lower power consumption due to reduced number of collisions.

A decision of selecting an appropriate PHY layer transmission parameter, such as spreading factor, channel frequency, and power is impacted by a number of factors. A couple of most important factors in this regard are: (i) channel condition, and (ii) distance of an ED from a gateway. It is a well-known fact that interference and data traffic load is not constant, hence the channel condition is variable with respect to time. Hence, there is an absolute need for a proactive, intelligent, and adaptive PHY layer transmission parameter adjustment algorithm for a LoRaWAN network. Hence, here we present a network-aware DRL framework for EDs' PHY layer parameters selection with the aim of maximizing PDR and lowering the power consumption.

The following are our major contributions:

- We have presented a DRL-based adaptive PHY layer transmission parameters selection algorithm for LoRa-based EDs.
- We perform performance evaluation of our algorithm under different environment settings and show that our proposed algorithm shows an improvement of more than 500% of PDR upon the learning based technique in some cases while being adaptive at the same time.

II. Background

A. LoRa Networks

LoRa utilizes the chirp spread spectrum (CSS) technique for encoding signals into chirp pulses spread over a wide spectrum. CSS enables long-range communication with robustness against interference and noise while keeping the data-rate low. LoRa allows the selection of different PHY layer transmission parameters (spreading factor, coding rate, bandwidth, frequency, and power) for each device. The values of these parameters affect communication range, data rate, resilience against interference, and a receiver's ability to decode the signal. In LoRa, a transceiver can select a bandwidth (BW) in the range 7.8 to 500 kHz and mostly a LoRa transceiver operates at 125 kHz, 250 kHz, or 500 kHz. Spreading factor (SF) defines the ratio between the symbol rate and the chirp rate. LoRa provides seven SF rates to choose from (SF6 to SF12). Coding rate (CR) defines the level of protection against interference. LoRa defines four coding rates: $\frac{4}{5}$, $\frac{4}{6}$, $\frac{4}{7}$, $\frac{4}{8}$. A LoRa radio can transmit between -4 to 20 dBm in 1 dB steps. However, due to hardware limitations, the mentioned range is mostly limited between 2 to 20 dBm.

TABLE I: General Comparison of LoRaDRL, LoRa-MAB, and LoRaSim

	Learning	Adaptive	Supports mobility	Average time to convergence
LoRaSim	✗	✗	✓	rule-based
LoRa-MAB	✓	∼	✗	100 khours (with no mobility)
LoRaDRL	✓	✓	✓	200 khours

B. Deep Reinforcement Learning

Reinforcement learning (RL) provides the ability to solve dynamic sequential decision-making problems. The conventional RL solutions have suffered from computational complexities due to the curse of dimensionality problem. Q-learning [6], an RL algorithm which involves learning using delayed rewards, suffers from the same problem as soon as the representation of the environment becomes complex. Mnih et al. [3] introduced DRL by proposing Deep Q-Networks (DQN), a combination of Deep Neural Network (DNN) and Q-learning as a solution to the stated problem. In the case of DQNs, the policy is represented by the DNN. Van et al. [5] pointed out that the DQNs may overestimate the Q-values and propose a new method called double deep Q-learning (DDQN). In DDQN, the action selection is proposed by an online network, but its value estimation for the update is done by using a target network. This not only helps estimating better Q-values but also increases the stability of learning.

C. Related Work

Bor et al. [2] proposed LoRaSim simulator for experimenting with different LoRa settings. This simulator provides an ability for dynamic PHY layer parameters selection where fixed subsets of the PHY layer parameter combinations are used to ensure collision avoidance. The only problem with their technique is that it suffers from the problems associated with a rule-based mechanism. Ta et al. [4] proposed the use of RL for dynamic PHY layer transmission parameters selection for LoRa-based EDs. They pointed out multiple issues with LoRaSim, for example, using perfectly orthogonal spreading factors. Based on their identified weakness in LoRaSim, they proposed another discrete event simulator named "LoRa-MAB". We identify multiple issues with LoRa-MAB: (i) LoRa-MAB, in terms of its computational complexity, is exponentially complex and hence not feasible for a large number of EDs; (ii) LoRaMAB does not account for the movement of EDs which makes it inapplicable in a network consisting of mobile EDs, such as health-care, smart vehicles etc. (iii) due to a missing specialized objective function, EDs have the option of choosing any of the available power levels without particularly focusing on saving power. As a solution to the above-stated issues, we propose an adaptive PHY layer transmission parameters selection algorithm based on DRL. A comparison of our approach (LoRaDRL) with prior work (LoRa-MAB and LoRaSim) is presented in Table I.

III. Proposed PHY Layer Transmission Parameters Selection Algorithm

To the best of the authors' knowledge, there is no DRL-based solution available in the literature for PHY layer transmission parameter adaption that assures minimalist collisions in a LoRa-based network.

A. Problem Formulation

We model the LoRa network with a total k LoRa EDs in a network, and with all EDs being within the range of a LoRaWAN gateway. The algorithm is centralized with a DDQN being run on the gateway. The gateway is not limited in hardware and power resources, hence is able to efficiently run a DDQN. The whole operation is formulated as a Markov decision process (MDP), where 's' denotes the state of the environment (allocated actions, distance from gateway), 'a' denotes the action (the combination of SF and power) proposed by the DDQN, and 'r' denotes the reward at a time-step. The goal of the agent is to propose action in order to minimize the collisions while keeping power usage as low as possible.

B. Reward/Cost Function

The proposed reward function takes into account the PDR, packet airtime, and power usage of an ED. The reward/cost function is given in Equation 1. In the equation, PDR_t and at_t represent PDR and airtime in seconds respectively at time instance t. In the case of the availability of multiple power levels, we change the reward function as given in Equation 2.

$$r_t = a * PDR_t - b * at_t \tag{1}$$

$$r_t = a * PDR_t - b * at_t + c * PWR_t \tag{2}$$

where,

$$PWR = \frac{PowerMax - PowerChosen}{PowerMax - PowerMin} \tag{3}$$

where a, b & c are the relative constants used to assign appropriate weights to PDR, at, and PWR. We have tested the following combinations of the constants: $(a = 1, b = 0.3)$, $(a = 1, b = 0.5)$, and $(a = 1, b = 0.3, c = 0.15)$. These constants act as hyper-parameters and can be chosen according to the requirements of the application. Hence in the reward function, we have proposed to actively penalize the learning agent until it is able to achieve a good PDR while keeping the power consumption as low as possible.

C. Proposed Algorithm

Algorithm 1 shows the workflow of the proposed algorithm. Major benefits of our proposed algorithm are:

1) **Adaptive Behaviour:** The ability of DDQN to continuously learn based on the current performance makes it adaptive to the changing environment hence always changing the policy in the favor of better available actions.

Algorithm 1 DRL in LoRa Networks - Learning Process

Input: Q-Network Structure
Output: Trained Q-Network

Initialize both the Target and Online Q-Networks
Initialize the memory (replay buffer)
while $ep \neq maxEpisodes$ **do**
  while $steps < maxEdCount$ **do**
    Initialize the LoRa Network
    Compute state of the Network s_t
    Feed the state to the DNN to get action a_t
    Taken action a_t at state s_t
    Simulate the environment
    Compute reward r_t and next state s_{t+1}
    Collect m data-points (s_t, a_t, s_{t+1}, r_t) using policy π and add it to the memory
    Sample mini-batch from memory
    Compute the change in values using target Q-network Q'_ϕ: $y_j = r_j + \gamma \max_{a'_j} Q_{\phi'}(s'_j, a'_j)$
    Update the Online Q-Network: $\phi \leftarrow \phi - \alpha \sum_j \frac{dQ_\phi(s_j,a_j)}{d\phi}(Q_\phi(s_j, a_j) - y_j)$
    if $steps > targetUpdateInterval$ **then**
      Update the Target Q-Network $\phi^{'}$
    end if
  end while
end while

TABLE II: Specifications of the DDQN

No. of Layers	2
No. of Neurons	[16, 16]
Activations	[ReLU, ReLU, Linear]
Learning Rate	0.0005
Memory Capacity	30000
Batch Size	128
Gamma for Q-Values	0.7
Initial Epsilon	1
Final Epsilon	0.05
Change in Epsilon	0.00005
Update Frequency	3000

2) **Mobility Support:** The learning is being performed on the gateway and is independent of the individual EDs and the model can handle mobile EDs.
3) **Computationally Efficient:** The algorithm uses a small DNN in DDQN hence requiring very few computational resources. Our algorithm runs on the gateway and does not put extra burden on the resource constrained EDs hence adds to the applicability of our algorithm in real scenarios.

IV. Performance Evaluation

In our experiments, we consider an environment of 100 LoRa EDs spread in a radius of 4500 meters with a single base-station at the center. We use a data frame size of 50 bytes. Typical IoT use cases generate small data packets, hence 50-byte frame size can represent a large number of IoT use cases. The data is being generated using a poison distribution with a mean rate (λ) of 4 minutes. The simulation time is set to 50 times of the mean rate. The bandwidth is fixed at 125 kHz for all EDs. Currently, we have considered a single channel. The specifications of the neural network are given in Table II.

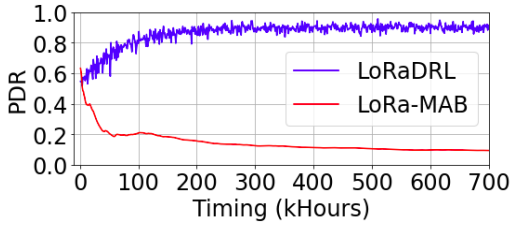


Fig. 1: PDR of LoRaDRL and LoRa-MAB in a LoRa network of 100 uniformly distributed mobile LoRa EDs with both the capture effect and the inter-sf collisions enabled. The mobility speed was set to 50±50 m/h. A sharp drop in the PDR of LoRa-MAB can be seen which shows its inability to learn in an environment comprising of mobile LoRa EDs.

A. Learning of the Proposed Algorithm

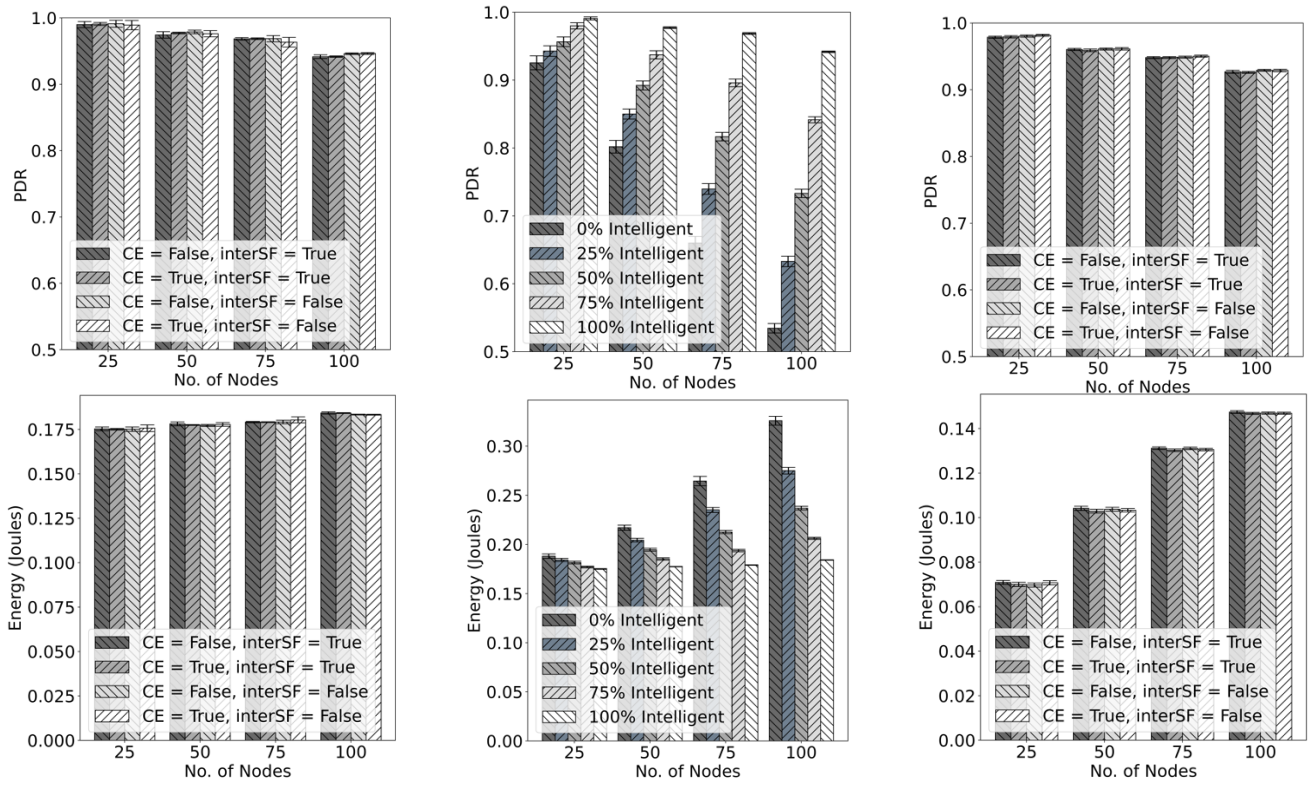
The PDR of our algorithm while learning is shown in Fig. 1. The figure shows that our agent can converge its learning in 200 khours, however LoRa-MAB is not able to learn a better PDR in case of an environment consisting of mobile LoRa EDs. EDs in the experimental setup are mobile and they follow the Gauss-Markov mobility model. An improvement of more than 500% can be seen in the PDR over LoRa-MAB. As the learning is independent of the EDs, so we propose the training of the model to be done in a simulation. The trained model can then be fine-tuned in a real environment. Due to the adaptive behavior of DRL, compared to LoRaSim and LoRa-MAB our model will be less susceptible to adversarial attacks which in the case of LoRa can be frequency jamming, etc.

B. Experiment 1: Performance evaluation using uniformly distributed EDs

Fig. 2(a) shows the performance of our algorithm in a field consisting of LoRa EDs distributed uniformly. We have tested with the capture effect (CE) and inter-SF collisions. The effect of these cannot be clearly seen in the provided graphs but will be noticeable in dense IoT networks. It can be seen that our model can achieve a PDR greater than 0.9 in a network containing 100 EDs in a single channel environment. Fig. 2(a) shows the average power usage per packet sent. It shows that our agent can achieve an average power usage of 0.185 Joule per packet with 100 EDs in the network which is an optimal power choice. The EDs have only a single power level to choose from, i.e., 14 db.

C. Experiment 2: Performance evaluation based on varying percentages of intelligent EDs

We compare the effect on overall performance in a network containing different percentages of intelligent EDs. We consider an ED whose decision is made by the DRL agent as an intelligent ED. We have tested with 0%, 25%, 50%, 75%, and 100%



(a) Experiment 1: Uniformly distributed EDs with a single available power level choice.

(b) Experiment 2: Performance of LoRa network containing different percentages of intelligent EDs.

(c) Experiment 3: Uniformly distributed EDs with multiple available power level choices.

Fig. 2: Depiction of performance of our proposed LoRaDRL. It can be seen that our proposed algorithm has achieved an optimal PDR while ensuring low power usage. In experiment 2, it is visible that the intelligent device percentage is directly proportional to the performance (high PDR & low power-usage). All results are reported with 95% confidence interval.

intelligent EDs. The EDs other than the intelligent ones choose a random parameter combination out of the available parameter combinations. We don't limit the parameter combination of any ED. All the available combinations are available to the ED to choose from. EDs have only a single power level to choose from, i.e., 14 db. It can be seen that the performance deteriorates in case of a reduction in the count of intelligent devices. When EDs choose a random parameter combination the packets either suffer collisions or they are lost. On the other hand, if an ED is intelligent, the parameters are chosen by the gateway based on the environment, hence fewer collisions. The obtained results have been shown in Fig. 2(b).

D. Experiment 3: Performance evaluation using multiple available power levels

In this experiment, we add multiple power levels as a choice [8, 11, 14] dbs for a power level to be used. In this case, the reward function given in Equation 2 is used. The results are shown in Fig. 2(c). It can be seen that our agent can achieve an optimal per packet power usage of 0.14 Joule which is much less than the average per packet power usage in the scenario of a single choice of power level, i.e., 0.18 Joule. Hence our agent can save power while ensuring the same PDR performance as in the case of one power level.

V. Conclusions

We have provided and tested the first deep reinforcement learning (DRL)-based approach for adaptive PHY layer parameters selection in dense LoRa networks that ensures fewer collisions and better performance than the existing state-of-the-art PHY layer parameter assignment algorithms. We show that our algorithm is not only adaptive and computationally efficient but is also able to support mobile end devices.

References

[1] L. Alliance, "LoRaWAN specification," *LoRa Alliance*, pp. 1–82, 2015.

[2] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?" in *Proceedings of the 19th International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, 2016, pp. 59–67.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[4] D.-T. Ta, K. Khawam, S. Lahoud, C. Adjih, and S. Martin, "LoRa-MAB: A flexible simulator for decentralized learning resource allocation in IoT networks," in *12th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, 2019, pp. 55–62.

[5] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[6] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.