Frequency Hopping Scheduling Algorithm in Green LoRaWAN: Reinforcement Learning Approach

Jui Mhatre[1], Ahyoung Lee[1*], and Hoseon Lee[2]
[1]Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA.
[2]Department of Electrical and Computer Engineering, Kennesaw State University, Marietta, GA 30060, USA
Email: [1]jmhatre1@students.kennesaw.edu, {[1]ahyoung.lee, and [2]hoseon.lee}@kennesaw.edu. ([*]Corresponding Author)

***Abstract*—Long Range Wide Area Network(LoRaWAN) is suitable for wide area sensor networks due to its low cost, long range, and low energy consumption. A device can transmit without interference if it chooses a unique channel, spread factor(SF), and transmission power(TP). In dense networks, the devices run out of unique choices, leading to interference and retransmissions. Eventually, the battery levels of devices drop faster. Also, the selection of transmission parameters affects the Time on Air(ToA) of the signal, thus increasing the energy usage of the end device. Higher values of spread factor, transmission power, coding rate, channel frequencies, and lower values of bandwidth increase energy consumption. Moreover, devices cannot maintain the same values of transmission parameters for long due to duty cycle restrictions. We use two techniques to deal with this situation: Frequency hopping and reinforcement learning. We proposed a scheduling algorithm, 'LoRa-DDPG-FHSS,' based on Frequency Hopping Spread Spectrum(FHSS) and Deep Deterministic Policy Gradient(DDPG) reinforcement learning for dense networks. It schedules device transmission parameters by selecting a unique(*channel*, *SF*, *TP*, *time slot*) to avoid interference and lower energy consumption. We performed simulations of three scheduling algorithms: FREE [1], LoRa-DDPG, and LoRa-DDPG-FHSS using the LoRaSim simulation tool. Our simulation analysis proves that LoRa-DDPG-FHSS improves energy efficiency by about 98% and Time on Air is reduced by 97% compared to FREE in a network with around 4000 devices.**

***Index Terms*—Reinforcement Learning, IoT, Deep Deterministic Policy Gradient, Frequency Hopping, Green LoRaWAN.**

I. Introduction

Discussion in "Worldwide Internet of Things(IoT) spending guide" [2] shows how fast IoT is spanning across the globe. IoT devices require a network that thrives on deficient Energy and has an extensive transmission range, making LoRa the best choice for IoT devices [3]. But LoRaWAN faces a few challenges, such as collision due to pure ALOHA-based MAC layer, network scalability, and its duty cycle restriction; energy consumption has become a significant concern with an increased number of devices in the network. Reducing this consumption for economic and environmental impacts is called green networking [4]. We would focus on reducing the energy usage in LoRaWAN to make it greener. LoRa uses an internationally reserved industrial, scientific, and medical(ISM) band with 868-870 MHz frequency and has about eight channels and about 1-10% of duty cycle, with each channel being used only for 36 sec in 1-hr. [5]. Increased network devices and duty-cycle increase waiting times, collision causing retransmission, and increased energy consumption.

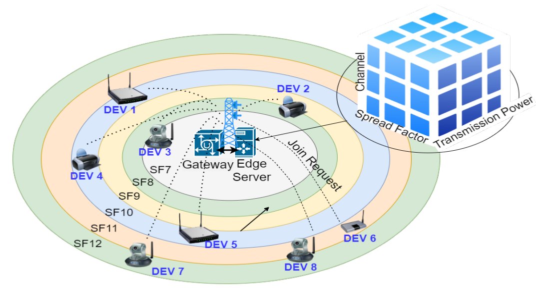


Fig. 1. Edge-enabled LoRaWAN with schedule generation for a particular time slot based on FHSS.

Other factors affecting energy consumption in LoRa are spread factor(SF) and transmission power(TP) [6]. 6 SFs range from SF7 to SF12, and these SFs impact data rate, Time on Air(ToA), battery life, and receiver sensitivity. Reference [7] shows the existence of global optima of TP for minimizing energy consumption achieved by configuring device transceivers. The optimal TP value results from adjusted RSSI(Received Signal Strength Indication) and SNR(Signal-to-noise Noise Ratio) received at the gateway, which is affected by indoor use or obstacles, antenna gain, and cable loss. Thus, antenna selection plays a vital role in LoRa communication by deciding the impact of the performance of the link, as well as the self-configuration of power received on an antenna [8]. Furthermore, there is a need for an edge computing technology with LoRaWAN for real-time monitoring and control solutions of massive IoT data overhead in hybrid networks, as discussed in [9] to work with 5G. This led us to design a battery-conserving solution to use edge-enabled LoRaWAN. IoT environment is highly dynamic, and its communication capabilities often result in sudden variations [10]. Some works [10]–[12] presented for the LoRa network propose using machine learning to improve the optimization results.

Intelligent Scheduling Algorithms for LoRa Minimizing Transmission Time: The Algorithm in reference [13] formulates the scheduling problem as the depreciation of transmission time. They formulated a minimization equation to find the total time slot, which gives the time duration of the frame or time required for all data to be transmitted. A scheduling algorithm, Fine-Grained Scheduling for Reliable and Energy-Efficient Data Collection(FREE) in reference [1], is meant for delay insensitive by scalable networks. The Algorithm aims to minimize energy consumption while following data cycle restrictions and eliminating collisions and grouping acknowledgments. It supports longer packets over shorter ones for energy efficiency. Deep Reinforcement learning-based algorithm for transmission power and spread factor selection for optimized energy efficiency for flying gateways in LoRa network is discussed in [14]. Underwater packet delivery ratio improvement is shown in [15] by optimizing spread factors. Another work for energy efficiency in [16] uses variable neighborhood search(VNS) and a minimum-cost spanning tree algorithm; it reduces low-power networks' implementation and maintenance costs. They use LoRa repeaters to increase the coverage of the LoRa network.

Scheduling Algorithms based on Frequency Hopping Mode for Reducing Collisions: Frequency-hopping spread spectrum(FHSS) transmits radio signals by rapidly changing the carrier frequency among many frequencies occupying a sizeable spectral band. Recent work in [5], [17], [18] shows Frequency hopping as a better solution for large-scale delay-insensitive IoT networks. Furthermore, they proved that path loss was majorly due to the loss of headers, and the capture effect can boost performance. Another reference for studying the scalability of the LoRa system can be found in [19]. They understood the limitation of transmitters that can be supported, which limits the capacity of the overall system.

Intelligent Scheduling Algorithms based on Frequency Hopping and Resource Allocation: LoRa-DRL proposed in [10], [12] is a deep reinforcement learning-based resource allocation and scheduling algorithm for dense and large-scale LoRa networks for learning transmission parameters. They aim to optimize packet delivery ratio(PDR) and energy consumption.

Based on our literature studies, we observed research issues, which are summarized as follows:

- Static scheduling algorithms for transmission time and energy minimization focus on a selection of SF but fail to consider channels, remaining data, or pseudo orthogonality of SF.
- When the number of registered devices exceeds the maximum scheduling limit in one cycle, the remaining devices may starve for their chance. Thus, it needs to do time slot-based scheduling as well.
- Reinforcement learning(RL) based algorithms for scheduling are an excellent option since scheduling needs to consider multiple factors in a dynamic environment. However, training and executing the model requires a lot of resources and incurs overhead on the Lora gateway.

We propose a scheduling strategy in an edge-enabled LoRa network to overcome its challenges by reducing the collision and improving the throughput and data rate while minimizing the energy consumption of devices and prolonging their battery life. Our scheduling strategy is summarized as follows:

- RL-based strategy calculation occurs at the edge server and lowers overhead at the LoRa gateway.
- We consider the four unique communication parameters(channel, SF, TP, and time slot) for scheduling, which gives an equal transmission opportunity to all registered devices and increases the overall data collection rate.
- We use Frequency hopping by further dividing channels and selecting a different one for every transmission.
- Our strategy optimally allocates SF, channel, TP, and time slot such that overall air time and, thus, energy consumption of the system is minimized.

The remainder of this paper is as follows. We present the system model of the scheduling strategy in Subsection II-A. We also discuss the simulation network and formulate the optimization problem in Subsection II-B. Further, we propose algorithms and discuss them in Section III. Section IV presents the simulation setup and discusses the simulation results and system analysis. Finally, the paper is concluded in Section V.

II. System Model and Problem Formulation

A. System Model

We propose using an edge-enabled LoRa network as shown in Fig.1 and employ a new reinforcement learning-based scheduling strategy. Since the LoRa gateway and edge server are located at the exact location, the energy consumed to offload computation to the edge is minimal. The network has energy IoT devices with periodic data transmission. These devices use uplink channels C to transmit the packets to the LoRa gateway using SF f at transmission time slot t with TP p. We use Frequency-hopping spread spectrum(FHSS) to increase network capacity with LoRa. Thus, we use network setups such as several channels, packet and fragment size, data rate, and coding rate similar to the long-range Frequency-hopping spread spectrum(LR-FHSS) [5]. The number of operating channel width(OCW) channels depends on the data rate. Since our setup uses DR8/9, we have 8 OCW channels C, 137 kHz each. To implement frequency hopping, each OCW channel is divided into 280 Occupied Bandwidth(OBW) channels c with a bandwidth of 488Hz each. Each device can use any subchannel c within channel C in the ISM band supported by LoRa with duty cycle restrictions d_c. Signals transmitted over a channel face interference if at least two of them select the same OBW channel c and the same SF f for transmission at the same time slot t. Thus, scheduling a packet transmission involves the selection of 4-tuple(c, f, p, t) that is(*OBW channel*, *SF*, *TP*, *time slot*) for each packet from all devices which has sent join-request to LoRa gateway such as to avoid collision and hence reduce energy consumption. Also, TP should be selected to ensure minimal energy consumption and acceptable signal strength. Moreover, as explained in [20], some spreading factors interfere, which should be considered while scheduling. The energy usage of any device is affected by the transmission parameter setup of the device.

1) *Spread Factor:* It determines the number of chips that form a symbol/chirp. A single increase in the Spreading Factor roughly doubles the duration of a chirp; fewer chirps per second are sent. Thus, the ToA of the transmitted signal increases with an increase in the spreading factor. ToA equals the time for which the transceiver is

transmitting the signal. Thus, the Energy consumed for transmitting the signal increases when ToA increases.

2) *Bandwidth:* With the increased bandwidth, the number of symbols transmitted in a given time increases. Thus, overall, ToA decreases and thus decreases the device's energy consumption.
3) *Central Frequency:* Higher central Frequency leads to an increase in ToA is shown in [21]. Thus increasing energy consumption.
4) *Coding Rate:* More chips per symbol are transmitted for higher coding rates, thus increasing ToA and energy consumption.
5) *Transmission Power:* It is signal strength generated by the transceiver. The generation of a signal with higher strength consumes higher energy. Thus, with an increase in transmission power, energy consumption increases.

In frequency hopping, all packets for each device with payload L and MAC header H are divided into fragments with interval 50 ms(RP2-1.0.2 LoRaWAN ®Regional Parameters) fragments(T_F) [22]. Each fragment is transmitted on a separate OBW, and then the device hops to the next OBW. Moreover, the duty cycle limitation of 3.9 kHz minimum separation, which is 8 OBWs(each of 488 Hz), should be maintained. The DDPG agent in the edge generates a schedule.

While transmitting the packet using LR-FHSS, we transmit three replicas of the header(for reliability) each of 233 ms header duration T_H and waiting time T_W of 2 bits transmission time. Reference [18] gives the Time on Air T_{air} of a packet for L bytes physical layer payload at LR-FHSS as,

$$T_{air} = 3 * T_H + T_W + 0.102\left\lceil \frac{L+2}{M} \right\rceil, \quad (1)$$

M=2 for DR8/DR10, and 0.102 ms is the payload duration. We define fragments as several bytes transmitted in one hop period T_F, which is 50 ms [18]. Thus, the number of fragments N_F for a given packet with payload L and two milliseconds guard time is,

$$N_F = \left\lceil \frac{0.102\left\lceil \frac{L+2}{M} \right\rceil}{T_F} \right\rceil. \quad (2)$$

Energy is consumed for each packet that is transmitted. We calculate the energy consumed to transmit all D data packets from a particular device using the equation defined in [1] as,

$$E = \sum_{i=0}^{D} (1 + R_i) * b_i * T_{air} * I * V, \quad (3)$$

where R_i is the number of times b_i packet is retransmitted, I and V denote current and voltage contributing to TP at the transceiver at the chip of IoT devices. We design our system such that it is collision-free. Thus, ideally, retransmission factor R=0. However, we can expect unregistered IoT devices to transmit and interfere, which causes retransmission.

Before the communication starts, every device must register using a join request and obtain its transmission parameters from the network server. MAC commands sets network parameters in each device. The device also sends information such as location and data buffer. Our learning model at the edge server generates and sends the schedule to all devices. This schedule contains the list of end devices and corresponding transmission parameters to be set during the transmission of each packet. Thus, this schedule gives separate transmission parameters for each packet D_i. LoRa gateway then collects the energy dissipation information from all devices and sends it to the edge server. The reward is calculated as the total energy consumed per device. The learning DDPG model learns an energy-optimized schedule in the next iteration. Other than interference and collision, packet loss is another factor to consider for retransmission. It is calculated using the received signal strength indicator(RSSI) and sensitivity. RSSI measures the power in the radio signal, which is an approximate value for signal strength received on an antenna. RSSI can be given as using [23],

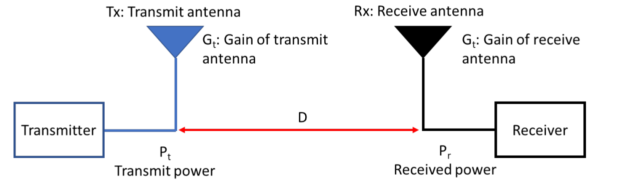


Fig. 2. Diagram illustrates the power and distance between an IoT device and the LoRa gateway separated by distance R. The LoRa gateway picks up the signal from one or more IoT devices and power from noise, which affects the signal-to-noise ratio(SNR).

$$RSSI = p - PL(d0) + 10 * \gamma + \log(d/d0) + \mu, \quad (4)$$

where p is TP, $PL(d0)$ is path loss at reference distance $d0$, γ is path loss exponent [24] and μ is unknown-but-bounded(UBB) noise with zero mean. In the RSSI equation, the path loss exponent is the average path loss for a given distance d concerning the reference distance $d0$, and γ is the path loss exponent. This path loss exponent can vary for different environments. If RSSI is lower than sensitivity [1], the packet is considered lost and must be retransmitted. However, the RSSI equation does not consider the antenna parameters and the noise level. Therefore, besides RSSI, the noise level should be regarded as to obtain the signal-to-noise ratio(SNR). Given the antenna specifications, the Friis formula provides another metric that can be used with the RSSI metric by considering the noise level and finding the SNR.

From the diagram in Fig. 2, the received power at a distance R away from the transmitter is given to be

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi R)^2}, \quad (5)$$

where G is the gain of the antenna, λ is the wavelength, and R is the distance between transmitting antennas. However, this equation only states the gain of the antennas, which is a function of the antenna efficiency and directivity. The following equations include the antenna efficiencies.

$$G_t = \eta_t D_t,\ G_r = \eta_r D_r, \quad (6)$$

where D is the directivity of the antenna η efficiency of the antenna. The receive antenna picks up both the signal and noise power at temperature(Ω), which can be expressed in terms of the bandwidth(BW).

$$P_n = K\Omega_s BW, \tag{7}$$

where P_n is receiver noise power, K is Boltzmann's constant [1.38 x 10^-23 (J/k)],Ω_s is system noise temperature [measured in kelvins], and BW is Receiver bandwidth in Hz. Now that we have the receive signal power and the receiver noise power, the signal-to-noise ratio, SNR, and its value in decibels can be defined as the following,

$$SNR = \frac{P_r}{P_n} = \frac{P_t G_t G_r \lambda^2}{(K\Omega_s BW)(4\pi R)^2}, \tag{8}$$

$$SNR(dB) = 10\log P_t G_t G_r \lambda^2 \text{-} 10\log(K\Omega_s BW) \text{-} 20\log(4\pi R)^2 \tag{9}$$

B. Problem Formulation

Our goal is to generate a device-wise transmission parameter scheduling sequence. It is a sequence of optimal values for three quantities: $OBW\ channel$, SF, and TP for all devices such that there is no interference. Each schedule is represented as a 3-dimensional matrix having eight channels, 6 SFs, and TP as three dimensions for each time slot. Each time slot is for a fragment duration time. Thus, we generate a schedule of transmission parameters for each data fragment for all devices. Thus, we can formulate this problem as follows,

$$\min \sum_{n=0}^{N} \sum_{b=0}^{B_n} r_b * T_{air} * I * V \tag{10}$$

s.t.

$$\sum_{n=0}^{N} \sum_{t=0}^{\tau} M[t][n] = \sum_{n=0}^{N} B_n \tag{11}$$

$$\frac{\sum_{t=0}^{\tau} M[t][n] * T_{air}}{\tau} \leq d \quad \forall n \in N, \tag{12}$$

where B_n denotes buffer size for device n and r_b is retransmission factor for packet b. Equation (10) is an energy minimization equation for all N devices. Equation (11) confirms that all B_n packets of each device are transmitted in total τ time slots, and each time slot t transmits $M[t][n]$ packets. LoRa, given by (12), satisfies duty cycle restriction where d denotes duty cycle percentage allowed(1%).

We formulate optimal energy for the scheduling problem as a Markov decision problem(MDP). It is a 4-tuple problem $(State(S), Action(A), Reward(R), TransitionMatrix(X))$.

- ***State(S):*** For each frame of a given time slot, the system observes a state which is input to it. The state is denoted as, $S = \{B, Loc\}$, where $B = B_1, B_2..B_N$ denotes the data in the buffer and Loc is their location.
- ***Action(A):*** It consist of actions taken to change the state. Action $A = A_1, A_2..A_N$ in our system denotes the scheduling decision for each device n represented as, $A_n = \{f, c, p, t\}$.
- ***Reward(R):*** Reward helps the DDPG model to learn and decide the action and update according to the reward. Our reward is based on energy consumption E found using (10) given by $R = -E$, where the negative sign shows that as energy reduces, reward increases and vice versa. Our DDPG model will learn that reward should increase, implying that energy consumption decreases.
- ***Transition Matrix(X):*** It denotes the initial state s_i transition to the next state s_{i+1} when action A is taken.

III. Proposed Algorithms

We propose two algorithms that are based on reinforcement learning. First is LoRa-DDPG, which generates a schedule of OCW channel C, SF f, and TP p by calculating rewards to minimize energy consumed by all devices. This Algorithm does not have time schedule, which may give rise to the collision. Our second approach, LoRa-DDPG-FHSS in Algorithm 1 uses frequency hopping and generates a time-based schedule by predicting the unique 4-tuple(c, f, p, t), where c is an OBW channel. Thus, we make a collision-free scheduling policy with minimal energy consumption. Collision testing at line 15 in Algorithm 1 involves checking $frequency\ collision$ and $SF\ collision$ for overlap in Frequency and SF. Collision increases energy consumption and thus reduces reward.

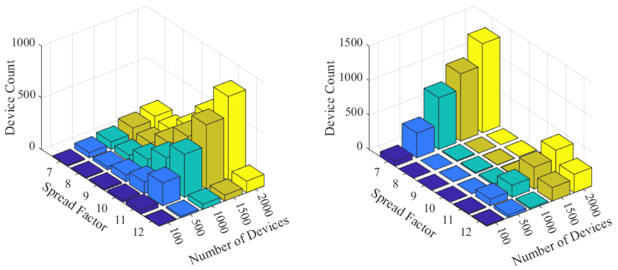
Algorithm 1 Algorithm for LoRa-DDPG-FHSS

Input: Initialize The LoRa network with N as a number of devices, locations, and packet sizes.
Output: Schedule for sending packets for all packets.

1: **for** $episodes$ in $MaxEpisodes$ **do**
2: [(C, f, p)] * N = DDPG(State)
3: $device\_sf\_txpow$= group devices based on SF, TP
4: $reward\_ep$ = 0
5: **for** $device\_list$ in $device\_sf\_txpow$ **do**
6: $t\_schedule$, $o\_schedule$ = generateSchedule()
7: $reward_f$ = Simulate transmission and calculateReward()
8: $reward\_ep$ += $reward_f$
9: **end for**
10: Learn DDPG network based on $reward\_ep$
11: **end for**
12: **calculateReward()**:
13: check overlap in the generated schedule
14: test collision in simulation
15: test path loss using RSSI and sensitivity
16: **if** collision **then**
17: $reward$ = MIN_INT
18: return $reward$
19: **end if**
20: Calculate Energy E, $reward$ = -1 * E
21: return $reward$
22: **generateSchedule()**: # adds time parameter to scheduling apart from(c, f, p) time=0
23: **for** fragment in device_list **do**
24: **for** device in device_list **do**
25: **if** obw == 280 **then**
26: $obw = 0, time += 1$
27: **end if**
28: $obw += 1$
29: $o\_schedule[device][fragment]$ = obw
30: $t\_schedule[device][fragment]$ = time
31: **end for**
32: **end for**

A. Unique schedule generation

Nodes are grouped based on SF and channel(collision as per LoRa-DDPG). LoRa-DDPG-FHSS uses method $generateSchedule()$ to assign unique OBW and time to each node. Every node gets a time schedule and an OBW schedule, each having length N_F as described in (2). In generated schedules, a maximum number of nodes transmit without overlapping in their OBWs for each time slot. Every OBW assigned to a given time slot is such that a minimum separation of 8 OBWs is maintained. Also, any OBW is not repeated for a given node in the next time slots if they consume all their duty cycles for that OBW. This confirms that the schedule of devices with the same SF is collision-free and without the same time slot and OBW during frequency hopping.



(a) FREE [1] (b) LoRa-DDPG-FHSS

Fig. 3. Comparison of SF distributions for the proposed LoRa-DDPG-FHSS algorithm and the existing FREE Algorithm using data size = 50 Bytes and an increasing number of devices ranging from 100 to 2000

B. Deep Deterministic Reinforcement Learning Algorithm

The proposed problem in (10) is an optimization problem where we need to optimize energy by adjusting channels, spread factors, and transmission power values. Transmission power is a continuous variable and makes its selection an exhaustive Energy. Thus, we can say that the problem we are trying to solve is NP-hard and quite complex. The reinforcement learning algorithm is a suitable solution for such optimization problems where the agent must perceive and interpret its environment and take action accordingly. Hence, reinforcement learning is our first choice to solve our problem. RL-based algorithms such as Q-learning(QL) and State–action–reward–state–action(SARSA) are suitable for limited action spaces like cart pole games. We resort to deep Q networks since our action space is much higher. There are(8 * 280 = 2240) OBW channels to select, seven spread factors, and 15 transmission powers. Selection in this space is quite expensive. Moreover, not all are discrete for the selection of transmission parameters; hence, we use the DDPG algorithm applied in our proposed Algorithm. 1.

IV. Performance Evaluation

A. Simulation Setup

We implement simulations LoRa-DDPG and LoRa-DDPG-FHSS using LoRaSim [19], which uses the Simpy library from Python. For machine learning, we use Python's Tensorflow libraries. We evaluate and compare results with FREE algorithm results using parameter setup in [1]. We use parameters for simulation discussed in Table I. Our learning agent gets input device information and learns the policy to generate optimal channel, spread factor, and transmission power. We collect results by increasing the number of devices ranging from 0 to 4000 and task sizes or data sizes from 10 to 50 bytes. We have extended this learning with the Frequency hopping strategy in LoRaSim to include OBW and time slot selection to avoid a collision. After every learning episode, we generate rewards as the negative value of energy consumption. The learning agent learns that optimal parameters generate maximum reward, meaning learning takes place to minimize energy. Thus, our Algorithm optimizes energy consumption.

TABLE I
Simulation Parameters

Parameter	Value
Number of IoT devices(N)	100-4000
Channels(C)	[867MHz, 867.3MHz, 867.5MHz, 867.7MHz, 867.9MHz, 868.1MHz, 868.3MHz, 868.5MHz]
Data size(B)	10-50B
OCW Bandwidth(OCW_BW)	137 KHz
OBW Bandwidth(OBW_BW)	488 Hz
OBW minimum separation	3.9 kHz
Coding Rate(CR)	1/3
Data Rate(DR)	8
Spread Factor(f)	[7,8...12]
Coding Rate(CR)	1/3
Payload fragment duration	50ms
Payload duration	102ms
Uplink channel duty cycle(d)	1%
Battery Capacity	1000 mAh
SNR limits, Receiver Sensitivities	[1]

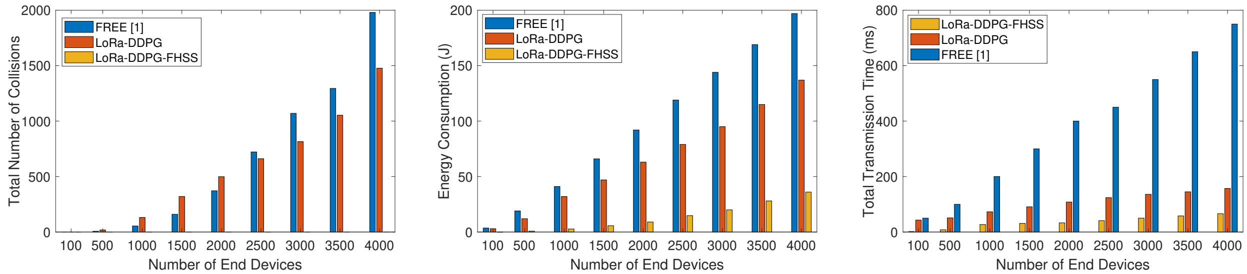
B. Discussion on the Simulation Results

We collect results for analysis and comparative study with existing algorithms based on our implementation. LoRa-DDPG uses only reinforcement learning for optimization, and LoRa-DDPG-FHSS uses reinforcement learning and Frequency hopping. We compare both their results with the heuristic approaches, FREE: Fine-Grained Scheduling for Reliable and Energy-Efficient Data Collection [1] and frequency hopping-based approach LR-FHSS [5]. Fig. 3 shows the evaluation of algorithms for SF distributions by increasing the number of devices: the existing heuristic FREE approach in Fig. 3(a), and the proposed LoRa-DDPG-FHSS in Fig. 3(b). When using LoRa-DDPG-FHSS, we focus on minimizing energy consumption. We observe that each device uses minimal SFs. SF is directly related to energy consumption. Hence, more devices use a lower SF, but if devices are far away, they need to minimize the SF among a higher spectrum. FREE fails to consider collection time when minimizing energy. In our proposed approach, in addition to packet loss validation, Frequency hopping minimizes energy consumption.

Fig. 4 compares the number of collisions, Energy consumption, and Transmission time when the number of devices increases in dense and scalable networks. Fig. 4(a) shows that the number of collisions observed in LoRa-DDPG-FHSS is the lowest and equals zero. But we see collisions in LoRa-DDPG and FREE algorithm. Fig. 4(b) shows lower energy consumption in both intelligent algorithms than FREE. Since the reward function of both DDPG-based algorithms is designed to minimize energy consumption, which is proven in evaluation results. Moreover, frequency hopping enables the full utilization of the network capacity and lower collision, reducing energy consumption. The total transmission time for all three algorithms is recorded in Fig. 4(c) and shows improved proposed approaches due to effective collision management.

V. Conclusion

We proposed an intelligent and energy-efficient frequency hopping-based scheduling algorithm called LoRa-DDPG-FHSS for edge-enabled LoRa networks. It improves energy efficiency(98%) and ToA(97%) as compared to existing ap-



(a) Collisions

(b) Energy Consumption

(c) Transmission Time

Fig. 4. Total number of collisions in Fig. 4(a), Total energy consumption in Fig. 4(b) and Total time in Fig. 4(c) required to transmit all packets using strategies(FREE [1], LoRa-DDPG, and LoRa-DDPG-FHSS in Algorithm 1) for increasing number of devices in the network under consideration.

proaches and makes it a greener LoRaWAN. The Frequency hopping technique deals with duty cycle restrictions and hops different frequencies for every fragment duration. The reinforcement learning-based Deep deterministic policy gradient algorithm helps to find optimal transmission parameters, including spread factor, transmission power, and central Frequency for all data fragments from all devices ready to transmit. We observe that unlike FREE, signal strength in our approach is maintained even when devices are far away from the gateway. LR-FHSS is designed without the scalability independent of increasing the transmitting packet rate by each end device; instead, it focuses on the overall network capacity increase provided by the statistical multiplexing of combining time and frequency diversity. As part of our future work, we plan to evaluate LoRa-DDPG-FHSS on the emulator and real LoRa network. This would help us test our Algorithm in real-world scenarios as well. We also plan to improve LoRa with Frequency hopping for communication with increased data to transmit, thus opening more use cases for LoRa.

References

[1] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch, “$free$—fine-grained scheduling for reliable and energy-efficient data collection in lorawan,” *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 669–683, 2019.

[2] M. Torchia, C. MacGillivray, A. Bisht, and A. Siviero, “Worldwide internet of things spending guide,” *Int Data Corp. https://www. idc. com/getdoc. jsp*, 2019.

[3] Semtech, “What Is LoRa,” https://www.semtech.com/lora/what-is-lora.

[4] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, “A survey of green networking research,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 3–20, 2010.

[5] G. Boquet, P. Tuset-Peiró, F. Adelantado, T. Watteyne, and X. Vilajosana, “Lr-fhss: Overview and performance analysis,” *arXiv preprint arXiv:2010.00491*, 2020.

[6] Y. Li, J. Yang, and J. Wang, “Dylora: Towards energy efficient dynamic lora transmission control,” in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 2312–2320.

[7] L.-T. Tu, A. Bradai, Y. Pousset, and A. I. Aravanis, “Energy efficiency analysis of lora networks,” *IEEE wireless communications letters*, vol. 10, no. 9, pp. 1881–1885, 2021.

[8] A. Gloria, C. Dionisio, G. Simões, and P. Sebastião, “Lora transmission power self con? guration for low power end devices,” in *2019 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC)*. IEEE, 2019, pp. 1–6.

[9] R. Sanchez-Iborra, J. Sánchez-Gómez, J. Santa, P. J. Fernández, and A. F. Skarmeta, “Ipv6 communications over lora for future iov services,” in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. IEEE, 2018, pp. 92–97.

[10] I. Ilahi, M. Usama, M. O. Farooq, M. U. Janjua, and J. Qadir, “Intelligent resource allocation in dense lora networks using deep reinforcement learning,” *arXiv preprint arXiv:2012.11867*, 2020.

[11] M. Chen, L. Mokdad, J. Ben-Othman, and J.-M. Fourneau, “Dynamic parameter allocation with reinforcement learning for lorawan,” *IEEE Internet of Things Journal*, 2023.

[12] I. Ilahi, M. Usama, M. O. Farooq, M. U. Janjua, and J. Qadir, “Loradrl: deep reinforcement learning based adaptive phy layer transmission parameters selection for lorawan,” in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*. IEEE, 2020, pp. 457–460.

[13] D. Zorbas, C. Caillouet, K. Abdelfadeel Hassan, and D. Pesch, “Optimal data collection time in lora networks—a time-slotted approach,” *Sensors*, vol. 21, no. 4, p. 1193, 2021.

[14] M. Jouhari, K. Ibrahimi, J. B. Othman, and E. M. Amhoud, “Deep reinforcement learning-based energy efficiency optimization for flying lora gateways,” *arXiv preprint arXiv:2302.05214*, 2023.

[15] S. S. Godkhindi, P. Reddy, and T. Prabhakar, “Link quality analysis of lora for internet of underwater things,” in *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 2023, pp. 222–224.

[16] M. A. Gava, H. R. O. Rocha, M. J. Faber, M. E. V. Segatto, H. Wörtche, and J. A. L. Silva, “Optimizing resources and increasing the coverage of internet-of-things (iot) networks: An approach based on lorawan,” *Sensors*, vol. 23, no. 3, p. 1239, 2023.

[17] M. El-Aasser, A. Gasser, M. Ashour, and T. Elshabrawy, “Performance analysis comparison between lora and frequency hopping-based lpwan,” in *2019 IEEE Global Conference on Internet of Things (GCIoT)*. IEEE, 2019, pp. 1–6.

[18] M. A. Ullah, K. Mikhaylov, and H. Alves, “Analysis and simulation of lorawan lr-fhss for direct-to-satellite scenario,” *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 548–552, 2021.

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

[20] C. Pham, A. Bounceur, L. Clavier, U. Noreen, and M. Ehsan, “Radio channel access challenges in lora low-power wide-area networks,” in *LPWAN Technologies for IoT and M2M Applications*. Elsevier, 2020, pp. 65–102.

[21] K. Johansson and C. Eklund, “A comparison of energy usage between lora 433mhz and lora 868mhz,” 2021.

[22] StudyCatch, “LoRaWAN LR-FHSS Time-on-Air (ToA) Calculator for Direct-to-Satellite Connectivity,” https://www.studycatch.com/2022/01/lorawan-lr-fhss-time-on-air-toa-calculator-for-direct-to-satellite-connectivity/, 2022.

[23] L. Zhang, B. Yang, and X. You, “Received signal strength indicator-based recursive set-membership localization with unknown transmit power and path loss exponent,” *IEEE Sensors Journal*, vol. 21, no. 22, pp. 26 175–26 185, 2021.

[24] J. Miranda, R. Abrishambaf, T. Gomes, P. Gonçalves, J. Cabral, A. Tavares, and J. Monteiro, “Path loss exponent analysis in wireless sensor networks: Experimental evaluation,” in *2013 11th IEEE international conference on industrial informatics (INDIN)*. IEEE, 2013, pp. 54–58.