DDPG_CAD: Intelligent Channel Activity Detection Scheduling in Massive IoT LoRaWAN

Jui Mhatre*, Ahyoung Lee†, Hoseon Lee‡, and Tu N. Nguyen*
*†Department of Computer Science, Kennesaw State University, Marietta GA 30060, USA.
‡Department of Electrical and Computer Engineering, Kennesaw State University, Marietta GA 30060, USA
Email: jmhatre1@students.kennesaw.edu*,{ahyoung.lee (Corresponding author)†, hoseon.lee‡, tu.nguyen*}@kennesaw.edu

***Abstract*—Beyond 5G and future 6G aim to address rising energy demands as the world becomes more interconnected. LoRaWAN network is an energy-efficient IoT solution, but frequent retransmissions can quickly deplete sensor batteries. Efficient traffic management and collision avoidance are crucial. Initially, LoRaWAN used ALOHA for multi-access, causing increased network collisions. The recent innovation of Channel Activity Detection (CAD) has emerged to tackle these issues. CAD enhances multi-access by sensing channel activity before transmission. Although an improvement, CAD is not foolproof. Our paper introduces enhancements to CAD through the Deep Deterministic Policy Gradient-based Algorithm (DDPG_CAD). To assess CAD functionality, we develop LoRaCAD, a dedicated simulator. We also conduct a thorough comparative analysis of scheduling strategies, considering energy efficiency, latency, and packet delivery ratio.**

***Index Terms*—LoRaWAN, deep deterministic policy gradient, channel activity detection, energy efficient.**

I. Introduction

5G and 6G networks aim to provide high-capacity, high-speed, ubiquitous, and green communication to improve coverage and energy efficiency. Some applications, such as smart agriculture, prioritize energy efficiency and extended coverage range over high-speed connectivity, where low-energy and long-range networks are crucial. LPWAN is a low-power, wide-area coverage technology. LoRaWAN, a major LPWAN tech, supports long-range and minimal energy consumption. It uses the ALOHA multiple access technique, leading to collisions and congestion. Duty cycle restrictions allow devices to transmit for 1-10% of the time, causing high latency. Massive IoT exacerbates these challenges due to increased network traffic.

Efficient channel sensing techniques like Channel Activity Detection (CAD) and Lightweight Carrier Sensing (LCS) can help overcome communication challenges. An optimal scheduling algorithm reduces collisions and selects transmission parameters for better spectrum usage and lower power consumption. In LoRaWAN, scheduling involves selecting transmission parameters that define signal characteristics. To improve results, CAD and LCS can be combined with scheduling strategies. A CAD-enabled simulator is needed to investigate channel-sensing-based scheduling strategies. Our study compared existing LoRa simulators as shown in Table I. Based on the studies of various simulators we observed

- Channel Activity Detection (CAD): Existing simulators did not support the CAD simulation.
- LoRa Gateway with energy module: Current simulators cannot furnish gateway details, such as battery level, to prevent battery exhaustion attacks [1].
- LoRa Gateway with load handling: The current simulators do not consider device scalability limitations.
- Integration of scheduling algorithms: Developing a simulator seamlessly incorporating reinforcement learning-based algorithms is essential, eliminating users' need for intricate network knowledge.

LoRaWAN uses an efficient channel sensing technique called Channel Activity Detection (CAD) to reduce energy consumption. However, CAD has limitations when operating on busy channels, leading to delays and potential collisions. A new method in [6] called LCS based on CAD has been introduced to enhance network scalability and reduce energy consumption. A CSMA-based multi-access technique is proposed in [7] to mitigate collisions for both short and long messages, especially during periods of high traffic, thereby lowering energy consumption. In [8], energy consumption is impressively reduced by 177%, presenting a mathematical energy efficiency model in multi-gateway LoRa. The model optimally allocates frequency channels, spread factors, and transmission power. Based on our literature review, limitations of existing simulators, and understanding of the requirements in LoRaWAN scheduling strategies, we propose,

- We developed a LoRa Simulator based on Channel Activity Detection (LoRaCAD) to assess the effectiveness of reinforcement learning (RL)-based scheduling strategies with channel sensing.
- We designed a scheduling strategy based on RL, specifically employing the Deep Deterministic Policy Gradient Algorithm with Channel Activity Detection (DDPG_CAD), which is proposed to optimize the selection of transmission parameters.
- DDPG_CAD also enhances the functioning of CAD by predicting the number of CAD repetitions.
- We assessed the performance of our algorithm by comparing the outcomes with studies in Table II.

The remainder of this paper is organized as follows. We give a network system model and formulate a research idea in II. We submit the structure of the solution in section III

TABLE I: Comparative Study of Simulators

Features	NS3 [2]	FloRa [3]	LoRaSim [4]	LoRa-MAB [5]	LoRaCAD
Language	C++, Python	C++	Python	Python	Python
Energy Model	✓	✓	✓	✓	✓
Adaptive Data Rate Support	✓	✓	x	x	✓
ACK Support	✓	✓	x	x	x
Imperfect SF	✓	✓	x	✓	✓
Signal Capture Effect	✓	✓	✓	✓	✓
Device Class	A	A	A	A	A
Multi-Gateway	✓	✓	✓	x	x
Channel Activity Detection	x	x	x	x	✓
Gateway Load	x	x	x	x	✓
Machine Learning Algorithm Integration	x	x	✓	✓	✓

and give implementation details of the scheduling algorithm DDPG_CAD and simulator LoRaCAD, followed by evaluation results in IV and conclusion V.

II. System Model and Problem Formulation

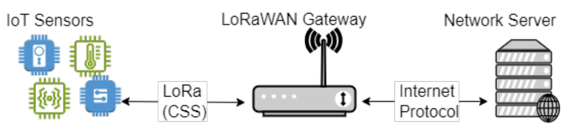


Fig. 1: Overview of LoRaWAN Architecture.

Our LoRaWAN network under consideration has N sensors, one Gateway, and one network server. It is a star topology network with all sensors connecting to the Gateway, which connects to the network server (Fig. 1). We focus our scope of research on communication between sensors and the LoRaWAN gateway. The US-based LoRaWAN has 72 uplink and eight downlink channels, also known as central frequency (CF), two bandwidths (BW), four coding rates (CR), 16 transmission powers (TP), and six spread factors (SF), [9] which count to 55,296 transmission settings. These are still limited settings in massive IoT if they transmit simultaneously. Moreover, LoRaWAN uses the ALOHA technique for multi-access. Proper scheduling of transmission parameters is required to optimize the network's data rates, airtime, and energy consumption. Thus, optimizing transmission parameters can improve LoRaWAN's energy efficiency. Another aspect is channel activity detection (CAD), which performs channel sensing for a preamble to check channel idleness.

CAD is an energy-efficient way to transmit sensor data by sensing channels before transmission. Before CAD operation, SF and BW are selected, and CADMode is enabled and inaccessible for $\frac{32}{BW}$ ms. Then for next $\frac{2^{SF}}{BW}$ ms actual sensing takes place and results are processed in $\frac{2^{SF}*BW}{1750e3}$ ms. The end of CAD processing is denoted by CADDone interrupt when we can check CADDetected interrupt for results about channel idleness [10]. If CADDetected is true, if the channel is busy, then the packet for which transmission was under consideration is dequeued from the buffer and appended to the end of the queue. If the CADDetected is false, then transmission starts at an immediate moment. We observe a few issues in this.

1) Collision for CADDetected = false: Fig. 2a shows a scenario of CAD when two or more devices with the same SF and BW perform cAD simultaneously on the same channel. Both devices detect that the channel is idle using CADDetection, start transmitting instantly, and end up in a collision. Massive IoT aggravates this issue.

2) Waiting time increase for CADDetected = true: Fig 2b shows a scenario when the channel is busy, and other devices perform CADDetection on this channel, they have to wait for transmission to be completed. Moreover, the packet trying to transmit is enqueued in the buffer and made to wait till others finish. This causes starvation for devices in busy networks.

Hence, we propose to design a new CAD strategy to improve multi-access using a scheduling algorithm using RL. Existing simulators (see Table I) lack CAD functionality and gateway load management, essential features for evaluating CAD-based algorithms and scheduling strategies in LoRaWAN networks.

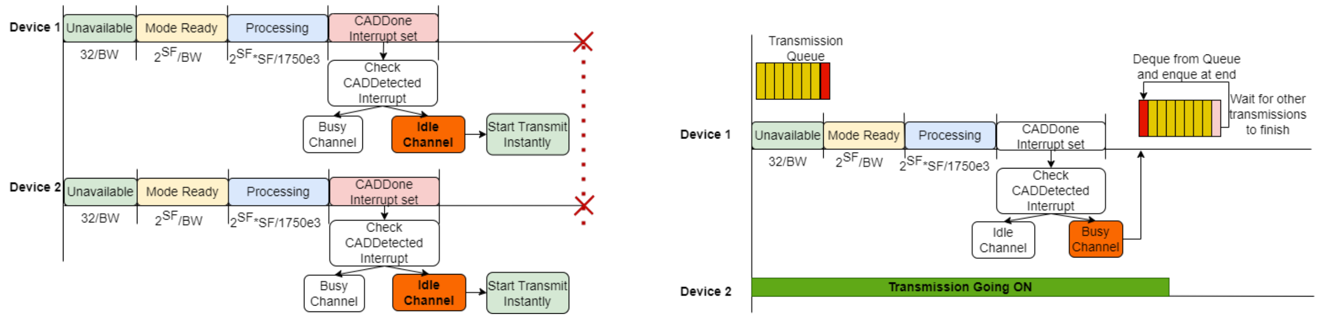
III. Proposed Solution

In this paper, we propose a twofold solution as discussed,

A. Scheduling Algorithm with Improved CAD: DDPG_CAD

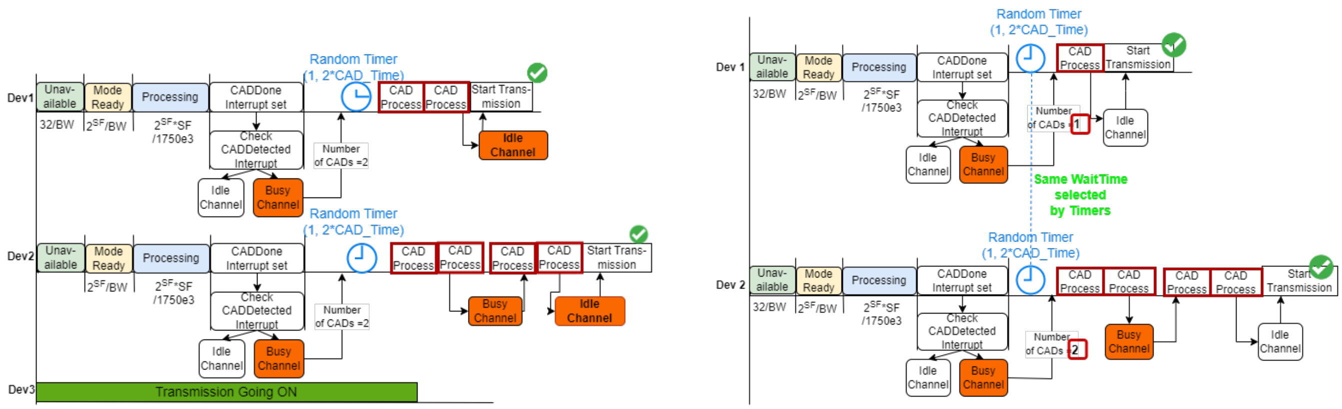
We have proposed a deep deterministic policy gradient RL (DDPG) algorithm that intelligently optimizes the selection of transmission parameters to avoid overlap and lower energy consumption by avoiding collisions. The state includes the location of devices, the data size they have to transmit, and the gateway load limit. The action consists of transmission parameters and CAD retries performed after the channel is found busy. The reward factor is $-1 * energy\ consumption$ since we aim to minimize the energy by maximizing the rewards. The transmission parameters are responsible for energy-efficient schedule generation. Input to the algorithm is a set of all sensor join requests with their information, such as locations or identifiers.

1) Solution to existing CAD issues: Two key issues are identified and solved in Fig. 2. To address these challenges, we introduce two enhancements to CAD using our algorithm: The number of CAD repetition predictions and the introduction of randomized transmission delays. Fig. 3(a) illustrates how the randomized timer introduces delays in CAD checks and enriches the likelihood of successful transmission even



(a) Idle Channel. (b) Busy Channel.

Fig. 2: **Issues with existing CAD in LoRaWAN:** Devices start transmitting instantly after identifying the idle channel and lead to collision (Fig. 2a). Delayed packet transmission if the channel is busy (Fig. 2b).



(a) Variation due to Randomized Timer. (b) Variation due to Number of CAD Repetitions.

Fig. 3: **Solution to issues with existing CAD in LoRaWAN:** Variation in Randomized Timer before CAD allows transmission devices identify other devices ready to transmit even if number of CAD repetitions are same (Fig. 3a). If not the same, then different CADs detect busy channels (Fig. 3b).

when the number of CAD repetitions remains the same. In Fig. 3(b), we demonstrate the effectiveness of predicting different numbers of CAD operations to prevent collisions. If a randomized timer initiates CAD processes simultaneously, unequal CAD checks ensure that devices with a higher number (e.g., device 2) can avoid collisions. If the channel is persistently busy, we limit the CAD operation to a defined number of repetitions. After that, the packet is enqueued.

2) Scheduling transmission parameters: Transmission parameter selection affects energy consumption in LoRaWAN. The SF defines a number of chips forming a symbol. This increases the transmission duration and the time the power amplifier continues to power the transmitter. BW specifies the number of symbols transmitted in a given time for an increase in BW, ToA, and energy consumption for the sensor decreases. Higher values of CF increase ToA [11]. Similarly, the coding range and transmission power can be scheduled to obtain different signal quality and control energy.

B. CAD-based Simulator: LoRaCAD

LoRaCAD has two primary functions: enabling CAD functionality in sensors and facilitating gateway load management. Three main elements are the - GW sensors and the environment. Within LoRaCAD, three-channel groups are available: 64 upstream channels (125KHz BW) and eight upstream and eight downstream channels (500KHz BW). Devices utilize these by selecting one channel as CF in transmission parameters. Other transmission parameters are scheduled according to the required signal and traffic conditions. The Network Server hosts the scheduling algorithm, offering the flexibility to integrate various scheduling algorithms and conduct comparative analyses. The Gateway module defines a limit of the device connections to the Gateway to manage the gateway load. CAD operations are managed within the transmission module using the parameter that allows the specification of the maximum number of CAD operations. LoRaCAD operates as a multithreaded simulator, each device functioning as an individual thread. When the thread starts executing, it requests a connection to the gateway. The connection is accepted if the gateway limit is not reached. The devices also notify the gateway of their locations. Gateway combines information about all devices and sends it to a network server to schedule the transmission parameters for all these devices. The transmission parameters are notified to devices using MAC commands. The device uses parameters for the packet that it wants to transmit. We show the method of transmission

TABLE II: State of the art Algorithms

Algorithm	Performance Parameter
ADR_MAX [12]	Energy and PDR
ADR_AVG [13]	Energy and PDR
NO_ADR [13]	Energy and PDR
ADR_Lite [14]	Energy and PDR
LP-MAB [15]	Energy and PDR
LoRa ADR+ [13]	Energy and PDR
LoRa ADR++ [16]	Energy and PDR
ASA [17]	Latency
DPST [17]	Latency and Energy

in Fig. 3. Any new scheduling strategy to be tested is stored on the network server and is referenced using the properties section.

IV. Performance Evaluation

We evaluate the DDPG_CAD algorithm on LoRaCAD and compare results with algorithms mentioned in Table II. We analyze average energy consumption per device, packet delivery ratio (PDR), and average latency. We set the environment using parameters as per the Table III. We evaluate algorithms mentioned in urban and suburban areas. The difference lies in network parameters altering the network traffic, path loss, distance from gateways, and range [13].

A. Discussion on the Simulation Results

Our evaluation is presented in two stages. First, we evaluate the performance of LoRaCAD with the existing simulator, LoRaSim. Second, we compare the existing algorithms with our DDPG_CAD on LoRaCAD. This shows the performance improvement of DDPG_CAD in the LoRa environment.

1) Comparison of LoRaCAD with LoRaSim: We assessed the performance of the LoRa-RL scheduling algorithm, as detailed in [18], on both LoRaSim and LoRaCAD platforms. Through these comparative evaluations, we aim to show the importance of having a CAD-enabled simulator to get a broader and more accurate picture of the network. It also helps us understand how channel sensing improves performance. Additionally, it helps evaluate ML/RL models with CAD-related predictions before emulating or deploying them in the network. LoRaSim uses packet retransmissions to handle losses, collisions, and lack of acknowledgment. LoRaCAD, on the other hand, employs carrier sensing through CAD to reduce collisions and retransmissions, leading to lower energy consumption. The disparity in energy consumption is evident in Fig. 4a due to the significantly lower energy expenditure in CAD operations compared to retransmissions. LoRaSim discards packets after a specified number of retransmissions, while LoRaCAD detects channel activity before transmission and initiates a CAD retry after a shorter backoff time. Our RL model predicts a maximum number of CAD retries. The channel is sensed before each retransmission until the packet is transmitted successfully or the retransmission limit is exhausted, increasing the probability of success. Hence, we observe in Fig. 4b that we gain a increase in PDR for LoRaCAD. We observe that increase in devices also increases the latency in Fig. 4c. LoRaCAD latency improves because CAD is faster than retransmission.

TABLE III: Simulation parameters.

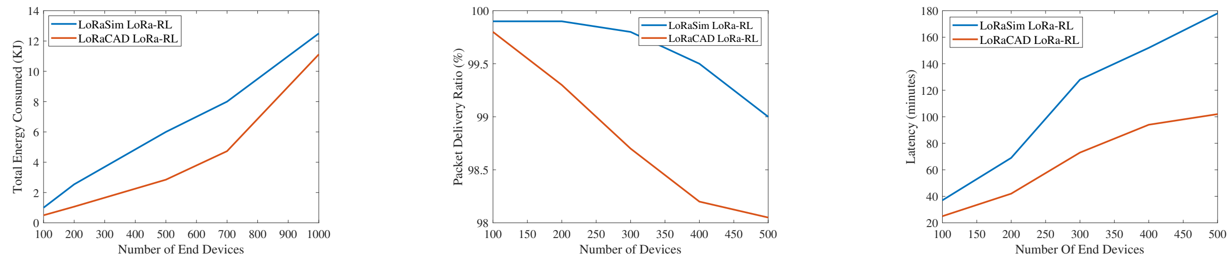
Parameter	Value
Bit Rate	Reference in [9]
Number of Device	1,000
Transmission Power	Reference in [9]
Gateway load Limit	120 connections
Signal to Noise Ratio	Reference in [19]
Reference Distance	40.0
path loss exponent	2.08
mean path-loss at d_0	127.41
CAD Backoff Timer	2ms
Network Area (Min)(X_0, Y_0)	(0,0)
Network Area (Max)(X_M, Y_M)	(1,000,1,000)
Gateway Location (L_{GX}, L_{GY})	(500,500)

2) Comparison of DDPG_CAD with existing scheduling strategies: We evaluate DDPG_CAD using urban and suburban scenarios with parameters in [13]. The network is denser in urban environments than in rural areas, causing higher collision probabilities. This emphasizes the significance of channel sensing-based scheduling. Conversely, in rural environments, the longer distances cause signal attenuation and loss. This necessitates an increased transmission power. Fig. 5a, Fig. 5b compares PDR (%) for various state-of-the-art algorithms to our proposed algorithm. CAD retries until an idle channel is found, or max retries are exhausted when the channel is busy. This process repeats before every retransmission. This reduces the possibility of collision and packet loss, increasing the PDR. Thus, our algorithm is a better solution for applications where data accuracy and reliable transmission are more critical.

Fig. 5c and Fig. 5d evaluate energy consumption for various algorithms to our proposed DDPG_CAD algorithm. The ADR++ consumes the lowest energy of all of them. More energy is consumed for channel sensing and retrying, which is less so in ADR-based algorithms. But DDPG_CAD consumes lower energy as compared to other existing algorithms. This makes our algorithm a better choice in unreliable networks to gain energy efficiency.

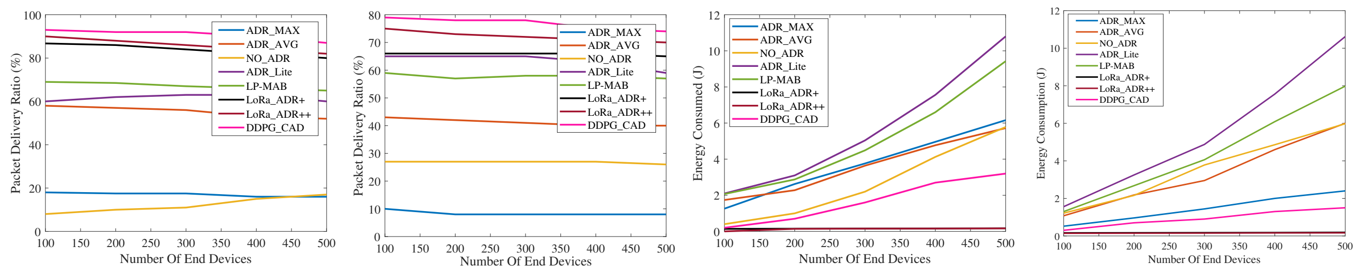
V. Conclusion

We have identified and addressed two pivotal challenges to improve the energy efficiency of sensors in a star topology for extensive IoT applications. The first is collisions in dense networks, while the second involves optimizing gateway load. To tackle these, we propose an innovative scheduling algorithm called the Deep Deterministic Policy Gradient Algorithm with Channel Activity Detection functionality (DDPG_CAD). This algorithm enhances Channel Activity Detection (CAD) by resolving existing CAD-related problems. We have developed LoRaCAD, a specialized simulator crafted to evaluate the performance of our energy-optimized scheduling algorithm. We compare energy and PDR metrics with state-of-the-art solutions in urban and rural environments. Our observations reveal a remarkable PDR of 87%, represent a substantial 6% advancement over the LoRa ADR++. Even with these



(a) Energy Comparison (b) PDR comparison (c) Latency Comparison

Fig. 4: Performance evaluation of LoRa-RL to transmit 10KB data using LoRaSim and LoRaCAD simulators.



(a) PDR: Urban (b) PDR: Rural (c) Energy: Urban (d) Energy: Rural

Fig. 5: Packet Delivery ratio and Energy Consumption for an increasing # of sensors. (Packet length = 20B)

enhancements in the context of massive IoT, our solution maintains lower energy consumption than most alternative algorithms. Energy consumption is at least 40% lower than most algorithms (except LoRa ADR++).

VI. Acknowledgement

We thank the anonymous reviewers for their suggestions and feedback. This research was partly supported by the US NSF Grants AMPS-2229073 and CNS-2103405.

References

[1] X. Yang, E. Karampatzakis, C. Doerr, and F. Kuipers, "Security vulnerabilities in lorawan," in *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2018, pp. 129–140.

[2] B. Reynders, Q. Wang, and S. Pollin, "A lorawan module for ns-3: implementation and evaluation," in *Proceedings of the 2018 Workshop on ns-3*, 2018, pp. 61–68.

[3] (2019) Flora site. [Online]. Available: https://flora.aalto.fi/

[4] (2019) Lorasim. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/regional-limitations-of-rf-use/

[5] 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 *2019 12th IFIP wireless and mobile networking conference (WMNC)*. IEEE, 2019, pp. 55–62.

[6] E. M. Rochester, A. M. Yousuf, B. Ousat, and M. Ghaderi, "Lightweight carrier sensing in lora: Implementation and performance evaluation," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[7] E. Sallum, N. Pereira, M. Alves, and M. Santos, "Improving quality-of-service in lora low-power wide-area networks through optimized radio resource management," *Journal of Sensor and Actuator Networks*, vol. 9, no. 1, p. 10, 2020.

[8] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, "Towards energy-fairness in lora networks," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 788–798.

[9] (2020) Lora alliance regeional parameters. [Online]. Available: https://lora-alliance.org/wp-content/uploads/2020/11/lorawan_regional_parameters_v1.0.2_final_1944_1.pdf

[10] (2023) Channel activity detection: Ensuring your lora® packets are sent. [Online]. Available: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/channel-activity-detection-ensuring-your-lora-packets-are-sent/how-to-ensure-your-lora-packets-are-sent-properly/

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

[12] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, "Energy consumption model for sensor nodes based on lora and lorawan," *Sensors*, vol. 18, no. 7, p. 2104, 2018.

[13] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of lora networks for dense iot deployments," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–9.

[14] R. Serati, B. Teymuri, N. A. Anagnostopoulos, and M. Rasti, "Adr-lite: A low-complexity adaptive data rate scheme for the lora network," in *2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2022, pp. 296–301.

[15] B. Teymuri, R. Serati, N. A. Anagnostopoulos, and M. Rasti, "Lp-mab: Improving the energy efficiency of lorawan using a reinforcement-learning-based adaptive configuration algorithm," *Sensors*, vol. 23, no. 4, p. 2363, 2023.

[16] Y. A. Al-Gumaei, N. Aslam, M. Aljaidi, A. Al-Saman, A. Alsarhan, and A. Y. Ashyap, "A novel approach to improve the adaptive-data-rate scheme for iot lorawan," *Electronics*, vol. 11, no. 21, p. 3521, 2022.

[17] Z. Ali, K. N. Qureshi, A. S. Al-Shamayleh, A. Akhunzada, A. Raza, and M. F. U. Butt, "Delay optimization in lorawan by employing adaptive scheduling algorithm with unsupervised learning," *IEEE Access*, vol. 11, pp. 2545–2556, 2023.

[18] J. Mhatre and A. Lee, "Dynamic reinforcement learning based scheduling for energy-efficient edge-enabled lorawan," in *2022 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. IEEE, 2022, pp. 412–413.

[19] (2019) The things network. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/rssi-and-snr/