LoRaWAN Physical Parameter Optimization Using Q learning

1st Piumika Karunanayake
Department of Communication Networks
University of Bremen, Germany
Bremen, Germany
piumikan@comnets.uni-bremen.de

2nd Andreas Könsgen
Department of Communication Networks
University of Bremen, Germany
Bremen, Germany
ajk@comnets.uni-bremen.de

3rd Thushara Weerawardane
Department of Electrical Electronic & Telecommunication Engineering
General Sir John Kotelawala Defence University
Ratmalana, Sri Lanka
tlw@kdu.ac.lk

4th Anna Förster
Department of Communication Networks
University of Bremen, Germany
Bremen, Germany
anna.foerster@uni-bremen.de

***Abstract*—LoRa / LoRaWAN is a cutting-edge technology that enables long-range and low-power communication for connecting sensor nodes, making it ideal for the Internet of Things (IoT) ecosystem. An adaptive data rate mechanism has already been incorporated into the LoRaWAN standard protocol and the network server monitors the signal-to-noise ratio to optimize the value of transmission power and the spreading factor. However, the legacy approach does not conduct any performance evaluation when changing the transmission parameters. In this work, we introduce a Q learning agent that dynamically fine-tunes the transmission parameters, i.e., transmission power, spreading factor, and code rate. The novel protocol aims to maintain the minimum required SNR to receive the signal and minimize the power consumption of end nodes while improving the packet delivery ratio. Our approach is validated through both simulation and real-world deployment. The simulation results reveal a significant increase in the packet delivery ratio, both for operation with one or multiple gateways, maintaining almost the same power consumption; the real-world implementation demonstrates the energy efficiency of the protocol.**

***Index Terms*—Parameter Optimization, Adaptivity, Q Learning, ADR, LoRaWAN**

I. Introduction

LoRaWAN, Long Range Wide Area Network protocol, is a wireless protocol for the Internet of Things (IoT), focusing on long-range transmission and low power consumption. Due to the low energy requirement, it is suitable for battery-powered nodes in remote areas. Meeting the requirements of IoT applications, nodes can be easily added and removed from the network. Using end-to-end encryption and device authentication, secure communication is ensured.

The performance of a protocol in terms of energy consumption, packet delivery rate, and delay is highly dependent on the environment [1]. Optimal operation of a protocol can be obtained by tuning parameters such as spreading factor (SF), transmission power (TP), and code rate (CR) for different environments. The gateway(s) can simultaneously decode signals modulated with different SF, as they have multichannel multi-modem transceivers [2]. The distribution of SF within the network minimizes interference in the network. Higher TP levels provide more extensive coverage, but increase interference and consume more power. Four different CRs are used for the robustness of the communication. However, selecting the optimal combination of all three parameters is not trivial.

In the standard LoRaWAN protocol, the Adaptive Data Rate (ADR) mechanism optimizes SF and TP for each node. The parameter determination is carried out by the network server for each node within a window of 20 consecutively received packets. The identified parameters are sent in acknowledgment packets.

The key contribution of the paper is the development of a novel optimization algorithm based on Q learning, which observes both the packet delivery ratio (PDR) and energy consumption per bit of each node. The proposed approach is initially validated using simulations. Furthermore, we design and evaluate a physical experiment in order to demonstrate the feasibility of our proposed algorithm in the real world.

The remainder of this paper is structured as follows. Section II discusses existing work carried out related to the optimization of protocol parameters. Section III presents a technological overview of the LoRaWAN protocol and Q learning. In section IV, the proposed model of the Q learning based protocol parameter algorithm is described in detail. The simulation setup and results are discussed in section V. The real-world implementation setup and results are presented in section VI. Finally, section VII concludes the paper, summarizing the key findings and future research possibilities.

II. Related Work

As mentioned in section III-B, the legacy ADR mechanism uses maximum SNR to optimize the transmission parameters. However, many other efforts exist to optimize the work of the standard ADR mechanism with various goals. The authors of [3] show improved performance by considering average

SNR for the ADR mechanism. Further, the authors propose assigning SF, based on the distance of the node to the gateway. However, to implement the mentioned algorithm, knowledge of the node's physical location is required. The assignment of SF among the nodes based on distance to the gateway has been proposed in [4]. The DyLoRa algorithm in [1] uses a prediction model to identify a better TP and SF to minimize the energy consumption of each node, sacrificing PDR.

Various algorithms, such as the STEPS, and LP-MAB, have been proposed to dynamically allocate transmission parameters [5], [6]. The authors of [7] have modified the ADR algorithm, incorporating a flexible link margin.

The authors of [8] and [9] use Q learning and deep Q learning, respectively, and show the results using simulations. The approaches of [10] and [8] have shown improvements in network performance. Increasing the number of nodes significantly degrades performance in a network, as proven by the work of [11], [12].

In summary, previous studies have suggested various algorithms to determine protocol parameters, considering factors such as average SNR, distance between node and gateway, and utilizing Q learning and Deep Learning algorithms. The state-of-the-art proposed algorithms improve performance and have been proven using simulations with a *small* number of nodes. Therefore, we identified a research gap in optimizing parameters considering a *large* number of nodes; also, to the best of our knowledge, no performance measurements to evaluate the performance of Q learning in a LoRaWAN network were taken so far in a real-world environment.

III. Technology Overview

A. Physical Layer and MAC Layer

The LoRa physical layer is defined in terms of modulation, frequency bands, data rates, SF, and TP. The devices operate in the unlicensed ISM (Industrial, Scientific, and Medical) bands, such as 868 MHz (Europe), 915 MHz (North America), and 433 MHz (Asia). Although the bandwidth can be set to 125 or 250 kHz, only the first option is widely used as a typical value. SF is the main modulation parameter, which can take values from 7 to 11 and decides the data rates and transmission range. The different TP levels 2, 5, 8, 11, 14, and 20 dBm affect the transmission range along with SF. The LoRa protocol allows for the use of different CRs 4/5, 4/6, 4/7, and 4/8. Although using a higher CR allows communication to be robust, it increases the packet on-air time, resulting in higher power consumption and an increase in the collision time.

LoRaWAN allows devices to be connected in an ad hoc manner. Pure ALOHA is used for channel access in the uplink. The duty cycle defines the duration of the allowed transmission time per cycle, and to facilitate connections for many nodes, the duty cycle of LoRaWAN is kept to as low as 1% [13].

B. Transmission Parameters and Adaptive Data Rate

To provide better performance, LoRaWAN provides adjustable parameter values for SF and TP. A one-step increase in SF and TP contributes 3 dB gain to SNR. The gain that can be obtained by changing CR is between 0.7-1.5 dBm [14]. However, the gain is significant once the SNR is below -4 dB [15]. Hence, CR changes should be encouraged in harsh channel conditions.

Adaptive Data Rate (ADR) is a technique that dynamically allocates the transmission parameters SF and TP used by the end devices according to the SNR at the gateway. The technique optimizes power consumption by adjusting the parameters. As ADR changes the data rate for each end device, the network capacity is also optimized [2].

In the ADR mechanism, if the difference between the maximum SNR value and the required SNR is greater than 3 dB within a 20 packet window, SF is reduced unless it is at the minimum level. If SF is in the minimum level, TP is reduced in steps of 3 dB. If the maximum SNR is lower than the required SNR, TP is increased. The spreading factor is increased by the node if it cannot hear from the Gateway. The newly determined parameter set is sent to the end device piggybacked in the acknowledge message. The ADR mechanism does not consider raising TP until SF is minimum, although there is a possibility that the combined adjustment may yield a better option.

C. Q learning

Q learning is a model-free reinforcement learning method that learns from the environment through interactions. The agent decides the next action based on the policy and the outcome of the interactions with the environment. The result of the actions is visible in the performance. The Q learner calculates a reward based on performance, and the Q value is calculated based on the reward for a particular state and action. Q learning decides between exploration and exploitation using a greedy algorithm. The learning parameters ϵ, α, and γ determine the proportion of exploration and exploitation, the step size of the algorithm, and the contribution of past experience to current decisions, respectively. Q learning can be used to optimize protocol parameters in a network with a higher number of nodes in a simulation environment, as well as the real world [16].

IV. Model of the Q learning based Adaptive Parameter Optimization

In the proposed model, the parameters to be adjusted, their initial values, and the set of permitted values are given as input. The parameter values are identified by the agent for each packet window by evaluating the performance within the window size. Once the values are identified, it sends the parameter values to the end nodes to update their transmission parameters.

Our work considers TP, SF, and CR to be adjusted using Q learning. Building on the improved results obtained by the authors of [3], the average SNR is considered to assess the quality of the link within the window size in our work.

For the three parameters, the range of values was selected as specified in Section III-A. Only in the case of TP, the highest level of 20 dBm was not included. As already mentioned in

Algorithm 1 Selection of Action

`int` $N \leftarrow \frac{SNR}{3}$
if $N = 0$ **then** $S \leftarrow true$
end if
if $\epsilon \leq 0.2$ && $N \geq 0$ **then** best parameters
else
  while $N < 0$ **do**
    $i \leftarrow (randomNumber \,\%\, 10)$
    if $i \geq 6$ && $SF\,!= max$ **then** $SF \leftarrow SF + 1$
    else
      if $TP\,!= max$ **then** $TP \leftarrow TP + 1$
      end if
    end if
    $N \leftarrow N + 1$
  end while
  while $N > 0$ **do**
    if $CR\,!= min$ **then** $CR \leftarrow CR - 1$
    end if
    $j \leftarrow (randomNumber \,\%\, 10)$
    if $j \geq 6$ && $SF\,!= min$ **then** $SF \leftarrow SF - 1$
    else
      if $TP\,!= min$ **then** $TP \leftarrow TP - 1$
      end if
    end if
    $N \leftarrow N - 1$
  end while
  if S && $(PDR \leq 100\%)$ **then**
    $k \leftarrow (randomNumber \,\%\, 3)$
    if $k == 0$ && $CR\,!= max$ **then** $CR \leftarrow CR + 1$
    end if
    if $k == 1$ && $SF\,!= max$ **then** $SF \leftarrow SF + 1$
    end if
    if $k == 2$ && $TP\,!= max$ **then** $TP \leftarrow TP + 1$
    end if
  end if
end if

Section III-B, the change in TP and SF by one step improves the link budget by 3 dB. However, the improvement gained by a step change in CR provides a gain less than 2 dBm [15].

The Q learning algorithm evaluates the performance in terms of the PDR and energy consumption for a successfully delivered packet. The learning parameters ϵ, α, and γ were set to 0.8, 0.2, and 0.3, respectively, after performing several simulations. The state of the system is defined using the TP, SF, and CR values. Therefore, there are 120 possible states. A transition from one state to another occurs based on the packet delivery rate, SNR, TP, and SF or ϵ. Algorithm 1 illustrates the performance evaluation and obtaining the most suitable parameters by QADR.

The reward is calculated using power consumption per effective bit rate and PDR. The effective bit rate (EBR) is calculated using the following equation 1:

$$EBR = \frac{4SF \cdot BW}{(4 + crbits) \cdot 2^{SF}} \tag{1}$$

where BW is the bandwidth used, which is 125 kHz. $crbits$ is the number of bits added for every 4 bits due to CR.

The goal is to obtain the best parameter set that provides the highest PDR with minimum energy consumption. Therefore, the reward is calculated considering EBR, TP, and PDR, which is calculated by the network server.

$$Reward = \frac{4SF \cdot BW \cdot PDR}{(4 + crbits) \cdot 2^{SF} \cdot tp} \tag{2}$$

where tp is the TP in milliwatts. The reward is maximum with values of 100% PDR, $SF = 7$, and $tp = 2$. Once the reward is calculated according to equation 2, the Q value is calculated according to equation 3

$$Q(S, A) \leftarrow Q(S, A) + \alpha \cdot (R + \gamma \cdot \max_a Q(S', a) - Q(S, A)). \tag{3}$$

The performance of Q learning based ADR (QADR) is measured and compared with ADR using the packet delivery rate and the energy consumption per successfully received packet (E). The latter value E is calculated according to equation 4.

$$E = \frac{\text{Total energy consumption of all the nodes}}{\text{Number of successfully received packets at the NS}} \tag{4}$$

If no control messages are received from the network server, the node increases SF after waiting a predefined period [3]. However, there is a possibility that TP is required to increase when SF is at the maximum value. Hence in our work, after waiting for the predefined time as said before, the end device increases SF or TP with a probability of 0.7 and 0.3, respectively. The probability values were determined after running several simulations.

V. Simulation Experiments

The performance of the proposed algorithm was tested using a simulator and a real-world setup. Subsections V-A and V-B discuss the simulation setup and the results obtained. The real-life experiment is described in Section VI.

A. Simulation Setup

OMNET++ with the FLoRa framework is used as the simulator. FLoRa is an open source simulator and supports LoRa gateways to receive multiple packets from nodes in different channels [17]. A network with 1000 randomly deployed nodes is considered within an area of 1000×1000 m, considering an urban environment. Each node transmits a packet every 1000 s.

Two cases are considered for simulation: in the first case, the network consists of one gateway to send packets to the network server, while in the second case, the network consists of two gateways located 350 m apart within the same area as in the network with one gateway. The initial TP is set to 14 dBm and CR to 4/5 on all nodes. The SF is randomly set to a value in the range specified in section III-B. The path loss model

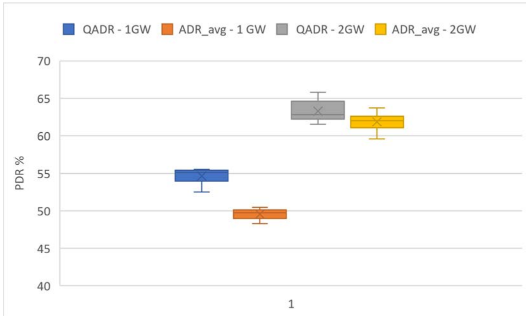

Fig. 1. Comparison of PDR

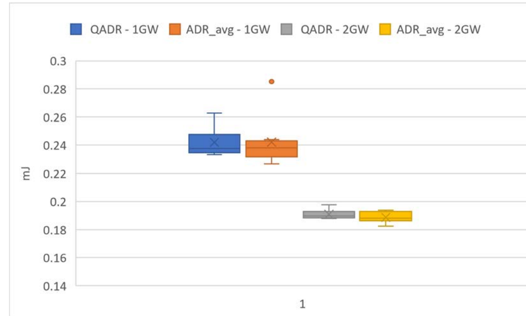

Fig. 2. Comparison of energy consumption per Rx Packet

is Log Normal Shadowing with sigma set to the default value 3.57. The simulation duration is set to 10 days.

In each case, 8 different networks with randomly deployed nodes were simulated and each network was simulated 10 times with a different random seed.

B. Simulation Results

Figure 1 illustrates the comparison of the packet delivery rate and the confidence interval of QADR and $\mathrm{ADR_{avg}}$ with one and two gateways, respectively. In the network with one gateway, the average improvement of QADR over $\mathrm{ADR_{avg}}$ is 5%, while the confidence interval width is less than 1%. The improvement in the network with two gateways is 1%, which is less than in the case of one gateway.

Figure 2 illustrates the comparison of energy consumption per packet of QADR and $\mathrm{ADR_{avg}}$ received successfully with one and two gateways, respectively. QADR consumes on average 2% more energy compared to $\mathrm{ADR_{avg}}$.

Further analysis is conducted to identify the distribution of SF, TP and CR within the network. Figures 3 and 4 illustrate the number of nodes in the network that have the same respective SF and TP values for both QADR and $\mathrm{ADR_{avg}}$. Figure 3 shows that QADR sets SF to different values in many nodes, whereas in $\mathrm{ADR_{avg}}$, mainly an SF of 9 is used. Figure 7 shows the distribution of $crbits$, within the QADR network. In the $\mathrm{ADR_{avg}}$ method, $crbits$ is not changed and set to 1. Although $crbits$ is not spread throughout the network, similar to SF or TP in QADR, different values are assigned. Hence, different CR values are also considered for the optimum parameter set.

The SF is distributed more in the network of two gateways compared to one gateway in both algorithms. Figure 5 shows the distribution of SF between nodes within the two gateway networks. QADR provides a greater distribution of SF between nodes compared to $\mathrm{ADR_{avg}}$. The TP distribution of the nodes in the same network is illustrated in Figure 6. The average TP is lower in QADR compared to $\mathrm{ADR_{avg}}$. Figure 8 represents the $crbits$ distribution of the network with two gateways. Many nodes use 1 as the $crbits$ compared to the network with one gateway.

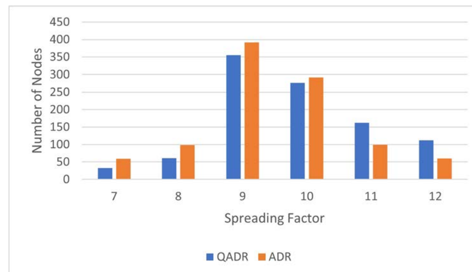

Fig. 3. SF distribution of the network with one gateway

VI. Real World Experiment

We implemented and tested a real-world setup in order to demonstrate the practical feasibility of our proposed Q learning algorithm.

A. Experiment Setup

QADR was implemented and evaluated using Heltec V3 nodes and a LoRaWAN gateway which connects to The Things

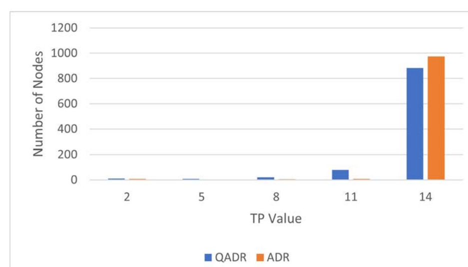

Fig. 4. TP distribution of the network with one gateway

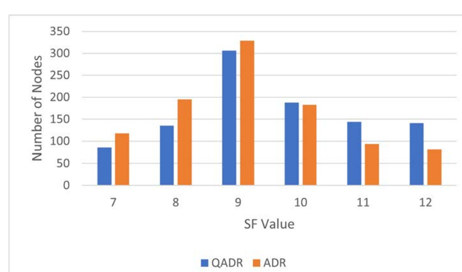

Fig. 5. SF distribution of the network with two gateways

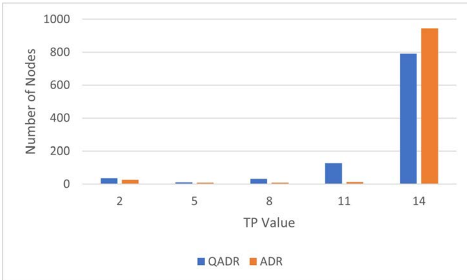

Fig. 6. TP distribution of the network with two gateways

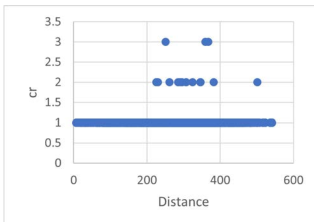

Fig. 7. *crbits* distribution in the network with one gateway

Network (TTN). Various connection metrics such as SNR, RSSI, TP, and SF data were obtained for each node and forwarded for data collection through MQTT messages. The QADR algorithm was implemented using Python, and parameter values to be updated were sent through acknowledgment messages to the nodes. The nodes were placed at 10 locations around the gateway, and transmitted a packet every 15 seconds. The parameters were set to the default values of the LoRaWAN standard. The bandwidth of 125 kHz, TP of 14 dBm, SF of 7, and a CR value of 4/5 were used as initial parameter values. After analyzing the simulation test results, we concluded that the impact of the CR change was less significant compared to the SF change and the TP change. Hence, the CR parameter was not selected as an adaptive parameter in the physical experiment. Therefore, Algorithm 1 was used without the steps incorporating CR.

For each node, the performance of QADR and legacy ADR using average SNR($\mathrm{ADR_{avg}}$) was measured after sending 301 packets.

To identify the significance of energy consumption for different SF and TP combinations, the battery capacity required to transmit 200 packets was measured for different

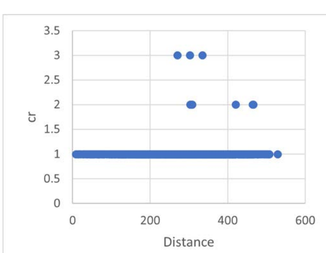

Fig. 8. *crbits* distribution in the network with two gateways

Location	Algorithm	PDR	Final SF	Final TXPwr
Location A	$\mathrm{ADR_{avg}}$	100	7	2
	QADR	100	7	2
Location B	$\mathrm{ADR_{avg}}$	100	7	14
	QADR	100	7	2
Location C	$\mathrm{ADR_{avg}}$	100	7	14
	QADR	100	7	2
Location D	$\mathrm{ADR_{avg}}$	100	7	14
	QADR	100	7	2
Location E	$\mathrm{ADR_{avg}}$	100	7	14
	QADR	100	7	14
Location F	$\mathrm{ADR_{avg}}$	92.6	7	14
	QADR	100	7	14
Location G	$\mathrm{ADR_{avg}}$	100	7	14
	QADR	100	7	14
Location H	$\mathrm{ADR_{avg}}$	100	7	14
	QADR	100	7	14
Location I	$\mathrm{ADR_{avg}}$	100	9	14
	QADR	100	7	14
Location J	$\mathrm{ADR_{avg}}$	96.6	7	14
	QADR	100	7	14

TABLE I

COMPARISON BETWEEN THE $\mathrm{ADR_{avg}}$ AND QADR PROTOCOLS

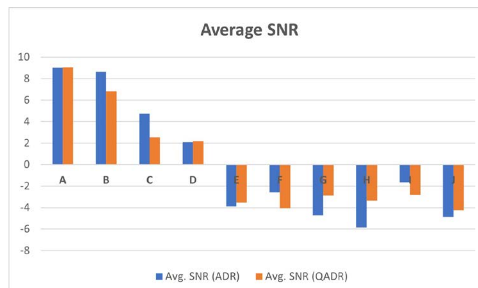

Fig. 9. Average SNR at the Gateway for different node locations

combinations of SF and TP. The battery charge was measured using the UNI-T UT658 USB tester.

B. Experiment Results

The connection metrics obtained in the experiment for the uplink packets are, average SNR, average RSSI, final SF after transmitting 301 packets, final TP after transmitting 301 packets and average error rate. Further, the percentage between the number of received packets at the gateway w.r.t. the number of transmitted packets was computed. The results are indicated for both protocols in Table I and in Figures 9 and 10.

The $\mathrm{ADR_{avg}}$ protocol transmits with the lowest power in location A, whereas in all other locations it transmits with the highest TP. The QADR protocol transmits with minimum power in four locations (A-D). For both protocols, from A to D locations, the PDR is 100%. Therefore, QADR can be observed to be a power-efficient protocol. Only in location 'I', $\mathrm{ADR_{avg}}$ increases its SF value to 9. However, QADR maintains SF at 7 and obtains 100% packet delivery rate. For locations E to J, the TP is kept at 14 dBm in both protocols.

Table II illustrates the battery charge required to transmit 200 packets for different combinations of SF and TP.

Fig. 10. Average Error Rate at the Gateway for different node locations

Test	SF	TP	Battery charge (mAh)
1	7	2	22
2	7	14	22
3	9	2	24
4	9	14	25
5	12	2	186
6	12	14	205

TABLE II
BATTERY CHARGE FOR DIFFERENT PARAMETER COMBINATIONS

VII. CONCLUSION AND OUTLOOK

In this work, we optimize the performance of LoRa transmissions by extending the legacy ADR scheme to adjust transmission parameters, incorporating machine learning. We considered three transmission parameters, TP, SF, and CR, to optimize while learning about the environment. Although higher transmission power improves the PDR, it adversely affects the network lifetime. Furthermore, increasing the spreading factor enhances the robustness of the channel while leading to higher energy consumption and on-air time. The use of QADR for adjusting the parameters improves performance in terms of PDR, keeping minimum energy consumption. As the CR's contribution is less than the SF and TP, the nodes try to maintain a 4/5 CR. Therefore, for real-world experiments, only the TP and SF are used for optimization.

In QADR, the nodes explore other possible combinations of parameters when the received SNR is between 0 and 3 dB. In contrast, in the legacy protocol and $\mathrm{ADR_{avg}}$, the parameters are unchanged. Therefore, the nodes can identify the best combination compared to the $\mathrm{ADR_{avg}}$ algorithm. The improvement in QADR is greater in the case of one gateway compared to two gateways, according to the simulation results. The SF values that individual nodes select are more distributed than in the case of $\mathrm{ADR_{avg}}$. Hence, more packets are correctly decoded at the receiver compared to $\mathrm{ADR_{avg}}$.

Allowing adjustments for TP as well as SF at the node in QADR provides more opportunities to reach the network server when control messages are not received, whereas $\mathrm{ADR_{avg}}$ only allows for increasing SF.

Real-world experiments show that increasing SF and TP requires higher power consumption. The QADR focuses on power optimization with higher PDR. Hence, it keeps minimum Tx power and minimum SF, maintaining a higher packet delivery rate compared to the legacy ADR.

An important direction for future work will be to extend protocol parameter optimization using Q learning for other communication protocols. Applying a similar approach as in the current work will optimize resource allocation as well as network performance according to the environmental conditions.

REFERENCES

[1] Y. Li, J. Yang, and J. Wang, "DyLoRa: Towards energy efficient dynamic LoRa transmission control," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. Toronto, ON, Canada: IEEE, 2020, pp. 2312–2320.

[2] C. Lehong, B. Isong, F. Lugayizi, and A. M. Abu-Mahfouz, "A survey of LoRaWAN adaptive data rate algorithms for possible optimization," in *2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*. Kimberley, South Africa: IEEE, 2020, pp. 1–9.

[3] 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*. Taipei, Taiwan: IEEE, 2018, pp. 1–9.

[4] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. Chania, Greece: IEEE, 2018, pp. 14–15.

[5] M. Chen, L. Mokdad, J. Ben-Othman, and J.-M. Fourneau, "Dynamic Parameter Allocation With Reinforcement Learning for LoRaWAN," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10 250–10 265, 2023.

[6] B. Teymuri, R. Serati, N. A. Anagnostopoulos, and M. Rasti, "LP-MAP: Improving the Energy Efficiency of LoRaWAN Using a Reinforcement-Learning-Based Adaptive Configuration Algorithm," *Sensors*, vol. 23, no. 4, p. 2363, 2023.

[7] G. G. M. de Jesus, R. D. Souza, C. Montez, and A. Hoeller, "LoRaWAN Adaptive Data Rate With Flexible Link Margin," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 6053–6061, 2021.

[8] R. Carvalho, F. Al-Tam, and N. Correia, "Q-learning ADR agent for LoRaWAN optimization," in *2021 IEEE international conference on industry 4.0, artificial intelligence, and communications technology (IAICT)*. Bandung, Indonesia: IEEE, 2021, pp. 104–108.

[9] 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)*. Sydney, NSW, Australia: IEEE, 2020, pp. 457–460.

[10] T. Fedullo, A. Morato, F. Tramarin, P. Bellagente, P. Ferrari, and E. Sisinni, "Adaptive LoRaWAN Transmission exploiting Reinforcement Learning: the Industrial Case," in *2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0 &IoT)*. Rome, Italy: IEEE, 2021, pp. 671–676.

[11] A. Lavric and V. Popa, "Performance evaluation of LoRaWAN communication scalability in large-scale wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 193–198, 2018.

[12] A. V. T. Bardram, M. D. Larsen, K. M. Malarski, M. N. Petersen, and S. Ruepp, "LoRaWan capacity simulation and field test in a harbour environment," in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, IEEE. Barcelona, Spain: IEEE, 2018, pp. 193–198.

[13] C. El Fehri, M. Kassab, S. Abdellatif, P. Berthou, and A. Belghith, "LoRa technology MAC layer operations and Research issues," *Procedia computer science*, vol. 130, pp. 1096–1101, 2018.

[14] J. Park, K. Park, H. Bae, and C.-K. Kim, "EARN: Enhanced ADR with coding rate adaptation in LoRaWAN," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 873–11 883, 2020.

[15] Q. Guo, F. Yang, and J. Wei, "Experimental evaluation of the packet reception performance of LoRa," *Sensors*, vol. 21, no. 4, p. 1071, 2021.

[16] P. N. Karunanayake, A. Könsgen, T. Weerawardane, and A. Förster, "Q learning based adaptive protocol parameters for WSNs," *Journal of Communications and Networks*, vol. 25, no. 1, pp. 76–87, 2022.

[17] A. I. Griva *et al.*, "LoRa-Based IoT Network Assessment in Rural and Urban Scenarios," *Sensors*, vol. 23, no. 3, p. 1695, 2023.