Handling Coexistence of LoRa with Other Networks through Embedded Reinforcement Learning

Sezana Fahmida
Wayne State University
Detroit, Michigan, USA
fahmida.sezana@wayne.edu

Venkata Prashant Modekurthy
University of Nevada, Las Vegas
Las Vegas, Nevada, USA
prashant.modekurthy@unlv.edu

Mahbubur Rahman
Queens College, City University of New York
New York, New York, USA
mdmahbubur.rahman@qc.cuny.edu

Abusayeed Saifullah
Wayne State University
Detroit, Michigan, USA
saifullah@wayne.edu

ABSTRACT

The rapid growth of various Low-Power Wide-Area Network (LPWAN) technologies in the limited spectrum brings forth the challenge of their coexistence. Today, LPWANs are not equipped to handle this impending challenge. It is difficult to employ sophisticated media access control protocol for low-power nodes. Coexistence handling for WiFi or traditional short-range wireless network will not work for LPWANs. Due to long range, their nodes can be subject to an unprecedented number of hidden nodes, requiring highly energy-efficient techniques to handle such coexistence. In this paper, we address the coexistence problem for LoRa, a leading LPWAN technology. To improve the performance of a LoRa network under coexistence with many independent networks, we propose the design of a novel *embedded learning agent* based on a lightweight reinforcement learning at LoRa nodes. This is done by developing a *Q-learning* framework while ensuring minimal memory and computation overhead at LoRa nodes. The framework exploits transmission acknowledgments as feedback from the network based on what a node makes transmission decisions. To our knowledge, this is the *first* Q-learning approach for handling coexistence of low-power networks. Considering various coexistence scenarios of a LoRa network, we evaluate our approach through experiments indoors and outdoors. The outdoor results show that our Q-learning approach on average achieves an improvement of 46% in packet reception rate while reducing energy consumption by 66% in a LoRa network. In indoor experiments, we have observed some coexistence scenarios where a current LoRa network loses all the packets while our approach enables 99% packet reception rate with up to 90% improvement in energy consumption.

IoTDI '23, May 09–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0037-8/23/05…$15.00
https://doi.org/10.1145/3576842.3582383

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; • **Networks → Wide area networks**; **Link-layer protocols**; *Network dynamics*; • **Computing methodologies →** *Q-learning*.

KEYWORDS

Internet-of-Things, IoT, Low Power Wide-Area Networks, LoRa, Reinforcement Learning, Q-learning

ACM Reference Format:
Sezana Fahmida, Venkata Prashant Modekurthy, Mahbubur Rahman, and Abusayeed Saifullah. 2023. Handling Coexistence of LoRa with Other Networks through Embedded Reinforcement Learning. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '23), May 09–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3576842.3582383

Sezana Fahmida, Venkata Prashant Modekurthy, Mahbubur Rahman, and Abusayeed Saifullah

1 INTRODUCTION

With its capability to enable low-power (milliwatts) wireless communication at low data rates (kbps) over long distances (kms), the *Low-Power Wide-Area Network (LPWAN)* technology is disrupting the Internet-of-Things (IoT) landscape. As such, LPWANs can enable many wide-area IoT applications such as smart cities, smart buildings, and smart metering where numerous low-power nodes directly forward information to a cloud-connected gateway [8, 9]. Due to their escalating demand in IoT applications, recent years have witnessed the emergence of numerous LPWAN technologies such as LoRa (Long-Range), SigFox, IQRF, RPMA , DASH7, Weightless-N/P, Telensa in the ISM band; EC-GSM-IoT, NB-IoT, LTE Cat M1, and 5G in the licensed cellular band; and SNOW [29] in the (unused) TV band.

The rapid growth of LPWANs in the limited spectrum brings forth the challenge of coexistence. The number of connected devices is expected to exceed 50 billion within a year [23]. With Comcast recently announcing to add LPWAN radios on set-top boxes, LPWANs will be ubiquitous in the US [22]. The coexistence problem will be severe in urban areas where the spectrum can be overly congested due to numerous independent networks. The immediate effect of such coexistence is degraded network performance in terms of throughput, latency, and energy consumption. Some networks or devices may even suffer from spectrum starvation. Repeated attempts to access the spectrum will drain their batteries. Studies show a collision probability of ≈ 1 if 1000 nodes of LoRa, SigFox, or IQRF coexist [12, 14]. Another study shows a throughput reduction of 75% when four LoRa networks coexist [37].

Today, LPWANs are not equipped to handle the impending challenge of coexistence. Their nodes (sensor nodes) have very low computation power, low memory, and limited energy typically supplied from small batteries. It is difficult to employ sophisticated media access control (MAC) protocol for low-power nodes. For example, LoRa MAC protocol, LoRaWAN, is a very simple and low-overhead protocol based on pure ALOHA, which even does not employ any sort of collision avoidance. Existing collision resolving approaches for LoRa are based on physical layer requiring a change in LoRa gateway and/or additional software-defined radio-based hardware, and work mostly for several packets collision [32, 35, 38–40]. Such reactive approaches cannot be generalized to handle the unprecedented number of collisions arising from numerous unknown coexisting devices/networks. Coexistence handling for WiFi, traditional short-range wireless sensor network (WSN), and Bluetooth [41, 42] will not work for LPWANs. In massive crowds of coexisting networks, the interference pattern can be hard to detect for an LPWAN node. Thus, a TDMA (time division multiple access) or CSMA (carrier sense multiple access) based approach will also fail. Due to long range, LPWAN nodes are subject to an unprecedented number of hidden nodes, requiring techniques that handle such coexistence while being highly energy-efficient.

In this paper, we study and propose to enable coexistence of LPWANs. We specifically address the coexistence problem for LoRa which is widely considered as an LPWAN leader [18]. It is commercially available all around the globe with more than 600 known use cases including smart cities, smart buildings, smart metering, and smart supply chain with over 50 million devices deployed on every inhabited continent [19]. Industry analyst IHS Market projects that 40% of all LPWAN connections will be based on LoRa in a few years [30]. Thus, many LoRa networks will have to coexist. Furthermore, it operates in the unlicensed ISM band which is limited and shared by various other LPWAN technologies such as SigFox, IQRF, RPMA, DASH7, Weightless-N/P, Telensa, and also by many other networks.

We develop a novel coexistence handling method for a LoRa network based on embedded learning on low-power nodes. As the wireless environment is largely unknown due to the coexistence of massive number of unknown networks, a *learning*-based approach becomes quite effective to take actions (e.g, transmit, sleep) according to the conditions. However, most machine learning algorithms are prohibitively computation-intensive and/or require considerable amount of memory which is again a big challenge for LPWAN nodes. To this end, we propose the design of a novel *embedded learning agent* at LoRa nodes based on a lightweight Reinforcement Learning (RL) to improve the performance of LoRa networks under coexistence with many independent networks. An RL agent (e.g., a node) is effective in an unknown environment where it learns through its experience. We specifically adopt *Q-learning*, a popular RL technique, which is well-suited for coexistence handling at the LoRa nodes as it entails relatively *low computation* requirement compared to other machine learning approaches. The proposed Q-learning framework exploits transmission acknowledgments as feedback from the network based on what an RL agent determines the actions to yield a successful transmission.

While Q-learning based approach was previously used in frame-based MAC protocols with time-synchronization to learn contention and collision with the nodes in the same network using a single channel [13, 15], we adopt Q-learning to handle coexistence with numerous unknown and uncoordinated LPWANs. Our proposed embedded Q-learning agent incorporates LoRa characteristics and works on top of traditional LoRaWAN with minimal overhead to the low-power nodes. To the best of our knowledge Q-learning has not yet been adopted to handle such coexistence. Specifically, the contributions of this paper are as follows. (1) We present the design of an embedded Q-learning agent for handling the coexistence of a LoRa network with other networks upon modeling the coexistence handling as a Markov Decision Process. The design ensures minimal memory and computation overhead at LoRa nodes. To the best of our knowledge, this is the **first** Q-learning approach for LPWAN and for handling coexistence for any low-power network. (2) Considering various coexistence scenarios of a LoRa network, we evaluate our approach through extensive physical experiments indoors and outdoors as well as through large-scale simulations in NS-3 [26]. The outdoor results show that our approach on average achieves an improvement of 46% in packet reception rate (PRR) while reducing energy consumption by 66% in a LoRa network. In indoor results, we have observed some coexistence scenarios where a current LoRa network loses all the packets while our approach enables 99% PRR with up to 90% improvement in energy consumption.

Section 2 presents an overview of LoRa and the system model. Section 3 presents related work. Section 4 demonstrates the performance of current LoRa network under coexistence through experiments. Section 5 presents the design of the proposed learning-based coexistence handling technique. Sections 6, 7, and 8 present the experimental results, simulations, and conclusion, respectively.

2 BACKGROUND AND SYSTEM MODEL

Here, we describe the necessary background for LoRa in Section 2.1 and the system model in Section 2.2

2.1 An Overview of LoRa

LoRa is a leading LPWAN physical layer technology that provides a communication range of 3-7 miles depending on the environment [3]. It ensures successful reception of packets at low signal-to-noise ratio (SNR). LoRa modulation is derived from *Chirp Spread Spectrum (CSS)*. CSS spreads the signal over the entire bandwidth, providing robustness to interference and enabling reception of packets at low/negative SNR. The modulated signal consists of symbols/chirps, whose frequency varies continuously over time. Information is encoded onto each chirp using multiple cyclic shifted chips. The number of chips in each symbol is controlled by the *spreading factor (SF)*. Specifically, SF is the ratio between symbol rate and chip rate, and is in the range [7,12]. It controls the data rate of transmission and hence the time on air and energy consumption. A higher SF causes lower data rate and higher energy consumption, and vice versa. Packets transmitted concurrently on different SFs on the same channel can be received successfully at a receiver.

Other configurable parameters for LoRa are channel, bandwidth, and coding rate. In the US, LoRa operates in the unlicensed ISM band (902-928MHz), and it defines 64 uplink channels with 125kHz bandwidth and 8 additional uplink channels with 500kHz bandwidth. For downlink, it defines 8 channels, each of bandwidth 500kHz. LoRa also supports different levels of forward error correction (FEC), called *coding rates* from $\frac{4}{5}$ to $\frac{4}{8}$. A higher coding rate provides higher reliability but increases the duration of each packet.

MAC protocol for LoRa is called LoRaWAN (LoRa Wide Area Network) which enables low-power and low data rate communication between thousands of end-devices/nodes. Numerous LoRa nodes are directly connected to one or more gateways forming a star topology. The gateways forward the data collected from the nodes to a central network server. The network server maintains the application requirement, security, and network parameters.

LoRaWAN supports three classes of operation, namely class A, B and C. In all classes, the nodes transmit using pure ALOHA. The differences in the classes is in the reception of packets from the gateway. Class A is the default class, where the nodes listen for packets for two short receive windows, immediately after transmission (mainly used for acknowledgments). In Class B, the gateway uses time-synchronized beacons and schedules receive windows for the nodes. In class C, the nodes continuously listen for packets.

2.2 System Model

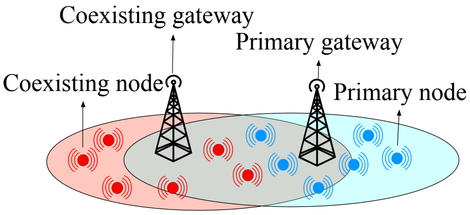


Figure 1: System Model

We consider a dense deployment where many independent LoRa networks coexist with multiple LPWANs in the same spectrum in Figure 1. Without loss of generality, we consider one LoRa network among them as the network of interest, called the *primary network*. The primary network consists of numerous nodes connected directly to one or more gateways. Each node is equipped with a single half-duplex radio and is energy and memory-constrained. The nodes sleep most of the time and wake up only to transmit a packet to the gateway. Each node in the network relies on acknowledgments (ACK) from the gateway to confirm the successful reception of a packet. The node retransmits the packet if an ACK is not received. The maximum number of retransmissions is configurable. The nodes periodically sense the environment and locally decide whether to report the data. Thus, there is a minimum inter-arrival time between packets. Note that the traffic in an LPWAN is mainly in uplink (nodes to the gateway). The nodes support multiple channels and can transmit on any of the available channels dictated by the link-layer protocol. The gateway is line-powered and connected to the Internet. It supports concurrent receptions of multiple packets on the same channel using different SF. The number of concurrent receptions is governed by the hardware specification.

We consider all other networks operating in the same spectrum as *coexisting* networks. The primary network does not possess any knowledge about the coexisting network and operates independently. Note that coexistence is different from jamming. In jamming, an attacker intentionally causes collision. On the contrary, in coexistence, limited spectrum causes collisions. Furthermore, the coexisting nodes are limited by the same constraints as the primary nodes, i.e., they are energy-constrained and do not possess the capability of transmitting continuously on all channels. The number of coexisting networks in the spectrum is large enough to cause significant performance degradation in the primary network. The communication technology of coexisting networks can be different from the primary network, however, they operate on the same spectrum.

3 RELATED WORK

Limited power budget of LPWAN nodes makes it difficult to adopt complex MAC protocol. Hence, SigFox and LoRa resort to ALOHA with no collision avoidance [34]. While such lightweight protocols provide energy efficiency, they are susceptible to interference and cannot handle coexistence with many networks. Several studies focused on analyzing the performance of LoRa under different coexistence scenarios [5, 25]. However, very few works considered handling the coexistence problem. A coexistence study through simulations revealed that using multiple gateways in distant places can marginally improve throughput at the cost of infrastructure [37].

Several works have studied the problem of resolving collided packets in LoRa through physical layer based approaches [32, 35, 38–40]. These works propose a reactive approach requiring a change in LoRa gateway and/or additional software-defined radio-based hardware. Besides, the effectiveness of such collision decoding approaches were demonstrated only for collisions with several packets. Such reactive approaches cannot be generalized to handle the unprecedented number of collisions arising from numerous unknown coexisting devices/networks. In contrast, we propose a link-layer based approach to handle massive coexistence which can be directly applied at the low-power LoRa nodes without requiring any physical layer change. The study in [27] uses the Poisson cluster process to model LoRa dense networks but does not propose coexistence handling. Recently, the work in [6] has studied the coexistence of LoRa and WiFi in the 2.4GHz band. However, the approach is applicable only when *the coexisting network uses WiFi*. In an uncoordinated environment with collisions from many *unknown networks*, such an approach will not work. Furthermore, the approach uses additional hardware for coexistence handling. In contrast, we propose a link-layer based approach that can be used with any existing LoRa network.

While there exists work on wireless coexistence considering WiFi, WSN, and Bluetooth (see surveys [42]), it will not work well for LPWANs due to their severe energy constraints. Due to large coverage domains, LPWAN devices can be subject to numerous hidden nodes. With their rapid growth in the limited spectrum, coexistence will be a severe problem and new energy efficient techniques must be developed to handle this problem. MAC approaches like ALOHA, TDMA or CSMA/CA may perform very poorly under severe coexistence. Due to the unprecedented number of hidden nodes in the network, a CSMA/CA based approach also becomes ineffective. A TDMA algorithm does not scale well with the large number of nodes. Furthermore, as the gateway is unaware of the coexisting networks, transmissions in any TDMA time slot are still subject to collisions. In contrast, the proposed learning based approach can dynamically infer other networks in the same spectrum at the nodes while being energy and computation efficient.

We adopt *Q-learning*, a widely adopted RL technique, which is well-suited for coexistence handling at LoRa nodes as it has *low computation* requirement compared to other machine learning approaches. It has been efficiently used in cognitive radios [43], and in WSN for routing [4], quality of service provisioning [45], and resource management [20, 24, 44, 47]. Effectiveness of Q-learning was studied through simulation for LPWAN [46]. Q-learning was also used in frame-based MAC protocols with time-synchronization to learn contention and collision with the nodes in the same network using a single channel [13, 15]. In contrast, we adopt Q-learning to handle coexistence with numerous unknown and uncoordinated LPWANs. Furthermore, many of the aforementioned approaches rely on computation-heavy deep reinforcement learning techniques involving deep neural networks which are not practical for resource-constrained LoRa nodes. Thus, we propose an embedded Q-learning agent which incorporates LoRa characteristic and works on top of traditional LoRaWAN with minimal overhead to the low-power nodes and evaluate through physical experiments. To the best of our knowledge Q-learning has not yet been adopted to handle coexistence of LoRa networks with many unknown networks.

4 PERFORMANCE OF A LORA NETWORK UNDER COEXISTENCE

LoRa adopts a novel physical layer technique to receive packets with a Signal-to-Noise ratio (SNR) 40 - 50dBm below the noise floor. Furthermore, LoRa nodes open two long receive windows of 1s for acknowledgments, limiting the packet rate from a node and improving the chances of successful reception.The effect of coexistence from other wireless technologies as well as other LoRa networks has not yet been studied empirically. Here, we show through physical experiments that the coexistence of other devices or networks in the limited band can severely affect LoRa's performance. Specifically, in this section, we experimentally answer the following questions. (1) Can other technologies interfere Lora? and (2) What is the performance of LoRa under coexistence from other LoRa networks?

4.1 Performance of a LoRa Network under Coexistence with Other Technologies

To test LoRa's performance under coexistence with other wireless technologies (of different physical layer), we run an experiment with Dragino SX1276 [17] mounted on Raspberry Pi 3 [28] as the LoRa node and a USRP (universal software radio peripheral) B210 device [36] as a coexisting node. We enable the LoRa node and coexisting node to transmit on the same channel in the US902-928MHz band and run two separate experiments. To represent different (than LoRa) technologies, in the first experiment, the coexisting node was using Amplitude Shift Keying (ASK) while, in the second one, it was using Frequency Shift Keying (FSK).

	Regular	Co-exist
exp1	100%	0%
exp2	100%	0%

Table 1: PRR under constant interference from other technologies

We show the Packet Reception Ratio (PRR) at the gateway under regular (i.e., without coexisting node) and coexistence scenario in Table 1, where the USRP was transmitting packets one after another without any gap to represent a severe coexistence scenario as if numerous coexisting nodes were communicating. On the other hand, the LoRa node was transmitting packets in 1 minute intervals using SF 10, coding rate $\frac{4}{5}$, and bandwidth of 125 kHz. In both experiments, LoRa PRR drops to 0% under coexistence.

To observe the performance of LoRaWAN's pseduo-random channel hopping against severe coexistence from other technologies, we run another experiment. We use two channels where the LoRa node was utilizing pseudo-random channel hopping, while the coexisting node was transmitting on a single channel only. We consider only ASK modulation based physical layer at the USRP device as the effect of FSK based physical layer was similar in the last experiment. All other settings were kept unchanged. In Table 2, we show the PRR, average retransmission attempts per packet, and total transmission energy consumption for 100 transmitted packets. We observe that LoRa nodes can deliver a packet to the gateway using retransmission on different channels. Specifically, each packet requires on average 2 transmissions for a successful reception. The increase in retransmissions also increases the energy consumption by 54%. Thus, pseduo-random channel hopping can improve the PRR, but at the cost of significantly higher energy consumption which is impractical for low-power nodes.

	Regular	Co-exist
PRR	100%	100%
Avg Attempts/packet	1	2.16
Energy (J)	3.58	7.63

Table 2: Results under coexistence with other technologies.

4.2 Performance of a LoRa Network under Coexistence with Other LoRa Networks

To analyze the effect of coexisting LoRa networks on a primary network, we ran another experiment with 2 independent LoRa networks coexisting in the same channel and same spreading factor. Each network consists of 5 nodes transmitting packets at 10 s intervals and uses separate gateways and network servers. The maximum number of retransmissions was set to 2 considering the network size. We transmitted 200 packets from each node and report the results for the primary network in Table 3. We observe that in this case, the primary network encounters packet failures and the PRR drops to 80% when the coexisting network is active. Furthermore, we see that the average transmission attempts also increases by 40% in the presence of a coexisting network, resulting in an increased Tx Energy consumption.

	Regular	Co-exist
PRR	94.3%	80.3%
Avg Attempts/packet	1.35	1.98
Energy (J)	9.06	11.308

Table 3: Results under coexistence with LoRa networks.

Thus, the impact of massive coexistence in LoRa networks is twofold. It hampers packet reception at the gateway resulting in degradation of the quality of service. Again, it results in high number of retransmissions at the energy-constrained nodes. Thus, coexistence handling is critical for a LoRa network. In the following section, we address this problem by developing a novel autonomous MAC protocol for LoRa that improves the PRR and reduces the energy consumption of each node. Note that the proposed MAC protocol is designed to handle severe coexistence scenarios. In case of negligible interference, the nodes should switch to regular LoRaWAN. The network manager can enable the proposed MAC protocol based on network conditions.

5 PROPOSED LEARNING-BASED APPROACH TO COEXISTENCE HANDLING

In the LPWAN coexistence scenario, the competing networks are uncoordinated. Therefore whether a transmission (Tx) will be successful or not depends on the present condition of the environment, irrespective of the past states. We show that this dynamic environment is well-suited to the Markov Decision Process (MDP) model, where the outcomes are determined by the *current* state of the environment. Thus, we first model LPWAN coexistence as an MDP.

Due to the substantial number of coexisting networks, the environment remains largely unknown to the nodes in the network. This facilitates the effectiveness of a *learning*-based approach. However, computation and memory intensive machine learning approaches are impractical in the low-power nodes. Thus, we propose a novel *embedded learning agent* based on RL to improve the performance of LoRa networks with coexistence of many independent networks without significant memory/computation overhead. Specifically, we utilize Q-learning, a popular RL approach with low computation requirements. A Q-learning agent learns the best action at every state through trials. Each action is quantified through Q-values, which are commonly stored in a lookup table, called *Q-table*. Due to the large number of nodes in a LoRa network, a centralized agent does not scale. Thus, we aim to embed the learning locally at the nodes.

We design an *embedded Q-learning agent* for LoRa networks to increase the number of received packets at the gateway, while incurring fewer retransmission attempts than traditional LoRa. The embedded agent is designed to utilize the characteristics of LoRaWAN. Specifically, LoRaWAN has the unique characteristic of concurrent reception across multiple *orthogonal* communication paths. Furthermore, the nodes can locally select the communication path at any time. The proposed embedded agent utilizes this by selecting the best communication paths based on observation and learning. Moreover, RL-agents interact with the environment by executing actions and receiving reward/penalty. Thus, the agent requires a feedback from the environment for learning. We achieve this by exploiting the acknowledgement mechanism of LoRaWAN. Each LoRa node opens two short receive windows after each uplink for downlink communication from the gateway. Thus, rewards and penalties are given based on successful reception of acknowledgements.

By evaluating each action through trials, the agent populates the local Q-table with a feasible memory overhead. The agent is able to perform significantly better under coexistence by exploiting the Q-table. Although the proposed Q-learning agent is designed based on LoRa's transmission characteristics, it can be easily adapted to other LPWANs as well. In the following sections, we first model LPWAN coexistence as an MDP. Based on the MDP, we present the design of an embedded Q-learning agent for LoRa which is feasible for energy-constrained low-power nodes.

5.1 LPWAN Coexistence as an MDP

An MDP is a discrete time state-transition framework where the outcomes of each decision are partly random and partly under the control of the decision maker [33]. It is formally represented as a 4-tuple (S, A, T, R), where S, A, T, and R are the set of states, set of actions, state-transition function and the set of rewards, respectively. Each primary node is an *agent* in the MDP. Note that, LoRa networks are multi-agent systems, however a multi-agent MDP is significantly difficult to solve. To remove the complexity of the problem, we consider other agents in the system as part of the environment and focus on single agent MDP formulation. When invoked, an agent in state s takes an action a, receives an immediate reward r and moves to the next state s' with a probability $\mathbb{P}[s'|s, a]$.

5.1.1 State Set. In our approach, the current state of the agent consists of the parameters of the communication medium and the state of the packet generated by the node. For a LoRa node, the medium consists of many communication paths. A communication path (CP) in LoRa consists of an uplink communication path (UCP) and a corresponding downlink communication path (DCP). A UCP is one element in the set of unique combinations of uplink channels and SFs on which a gateway can concurrently receive packets, while a DCP is a combination of a downlink channel and SF on which the node receives the ACK from the gateway. For LoRa, the outcome of packet transmission on one CP is independent of other CP.

When a packet is generated, it is in a *ready-to-transmit* state. If the node has decided to delay the packet, the packet is in a *sleep* state. If the packet has been delivered successfully to the gateway and an ACK is received, it moves to the *delivered* state. Note that if an ACK is not received, then the packet reverts to the *ready-to-transmit state.* Thus, each state $s \in S$ is represented as a 5-tuple $(c_i, f_i, c_j, f_j, \delta)$ where F, C_{up} and C_{down} represent the sets of SF, uplink and downlink channels, respectively; $c_i \in C_{\text{up}}$ represents the uplink channel, $c_j \in C_{\text{down}}$ represents the downlink channel, $f_i \in F$ represents the SF used for uplink, $f_j \in F$ represents the SF used for ACK, and δ represents whether the packet is in *ready-to-transmit*, *sleep* or *delivered* state. In LoRaWAN, the downlink channel for the ACK is decided by the gateway based on the uplink channel used by the node. Thus, the state set size is given by $3.|C_{\text{up}}|.|F|$.

5.1.2 Action Set. The agent is invoked when a packet has been generated and when a retransmission has been requested from the application layer. The possible actions for the agent are *transmit* and *sleep*. The agent can choose any available channel, $c_i \in C_{\text{up}}$ and SF, $f_i \in F$ to transmit. If the agent chooses to sleep, the sleep interval τ_i is chosen by the agent, along with the channel c_i and SF f_i. The agent transmits the packet after the interval τ_i on the chosen channel c_i and SF f_i. Thus, the set of actions is:

$$A = \{\textit{Transmit}\ (c_i, f_i),\ \textit{Sleep}\ (c_i, f_i, \tau_i)\ \}$$

It is important to bound the time a node can sleep and delay its transmission. An unbounded delay can halt the transmission for too long, seriously affecting the quality of service for many applications. To avoid this, we use a maximum delay, D, which denotes the maximum time a packet can be delayed in the *sleep* state.

The maximum delay D is divided into $\frac{D}{\phi}$ discrete transmission instants, where ϕ is the minimum interval between two consecutive transmission instants. A node can choose to sleep for a time $\tau_i = \eta\phi$, where η is an integer value in the range $[0, \frac{D}{\phi}]$. The value of the parameters ϕ and D can be decided by the upper layers based on application requirement and device memory constraints. The action set size is: $|C_{\text{up}}| \cdot |F| \cdot |\frac{D}{\phi}|$. The value of ϕ and D are the parameters that control the size of the action set. A high value of ϕ reduces the action set size, but limits the possible transmission instants for the agent. Thus, the agent is not able to fully explore the availability of the spectrum which can affect its performance. In contrast, a very low value of ϕ results in a high memory and computation overhead. Each agent in the system may select a different value of D and ϕ, as we consider the other agents present in the system as part of the environment. Thus, we do not require the nodes in the network to be time-synchronized.

5.1.3 State Transition Function. The dynamics of the environment is modeled through the state transition function, T. It defines the next state for each state and action pair with some probability. The function is given by $T : S \times A \longrightarrow \mathbb{P}(S)$ where an element of $\mathbb{P}(S)$ is a probability distribution over the set of states. Thus, $\mathbb{P}(S) = \{\mathbb{P}[s'|s, a] \ \forall s \in S, \ a \in A\}$, where s' is the next state. The next state of the agent depends on the immediately previous state and action (transmit/sleep). Thus, $\mathbb{P}[s'|s, a] = \mathbb{P}[s'|s_0, \cdots s, a_0, \cdots a]$ satisfies the *Markov Property.* Note that, while the agent can control the parameters (channel or SFs) of the communication path, the outcome of the packet transmission is unknown to it. Thus, the state transition function T for the MDP is unknown to the agent.

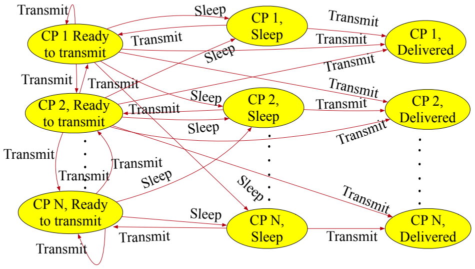


Figure 2: State transition diagram.

5.1.4 Reward Function Formulation. For every action a taken in state s, the agent receives an immediate reward r and incurs a cost $c(a)$ for taking the action. Then its total reward is given by: $r(s, a) = r + c(a)$

Considering both energy consumption and PRR, we assign some numerical values indicating rewards and costs. For example, we consider 1 unit of cost for a transmission attempt and 2 units of immediate reward if a corresponding ack is received at the node indicating a successful packet reception. Thus, a failed transmission will incur a cost of 1 unit. The reward function is given as follows:

$$r(s, a) = \begin{cases} 2 - 1 = 1 & \text{if Tx attempted; ACK received} \\ 0 - 1 = -1 & \text{if Tx attempted; ACK not received} \\ 0 & \text{if Tx not attempted} \end{cases}$$

A *policy* π is a mapping from states to actions. An agent's objective is to find a policy π^* maximizing it's total reward. Figure 2 shows the state transition diagram for the MDP.

5.2 Embedded Q-learning in LoRa

We propose to adopt a lightweight machine learning approach to find a feasible policy locally at the node for the LPWAN coexistence MDP. Specifically, we adopt Q-Learning that enables an agent to learn by interacting with its environment [33]. Q-Learning is a model-free RL algorithm which does not require the state transition probabilities to be known in advance. Nevertheless, it will learn to take the best actions that maximizes its long-term rewards by using its own experience. Furthermore, it is suitable for the LoRa coexistence scenario because it has low computation requirements than other machine learning approaches. To the best of our knowledge, this is the **first RL approach** for handling coexistence for any low-power network.

5.2.1 Rationale for Q-learning in LoRa. A centralized Q-learning algorithm is not scalable since it introduces significant communication overhead between the gateway and the thousands of nodes. Thus, we use an autonomous approach where each node uses an embedded Q-learning agent. Q-learning has been shown to converge to a reasonably good (which may not be optimal) solution even in a multiagent environment [10]. Thus, our goal is to design a lightweight agent capable of converging to a policy in a feasible time which can improve the performance of a LoRa network (in terms of PRR and energy consumption) in a dense coexistence scenario.

Finally, LoRa devices are extremely low-power and do not possess high computation power. Although Q-learning has a low computation requirement compared to other RL approaches, we have to be cautious in designing the Q-learning agent. The memory requirement of a Q-learning agent is dependent on the size of the actions and state space. In the case of a LoRa network under massive coexistence, each transmission at a different time can produce different outcome, and thus the action and state space becomes extremely large. To this end, in our design, we divide the action space in discrete time intervals to make the embedded agent memory and computation efficient. Specifically, the time τ_i in the sleep action is divided into $\frac{D}{\phi}$ steps. Note that our goal is to utilize the transmit and sleep actions to learn the best channel, SF and transmission time to maximize the packet reception rate at the gateway. While LoRaWAN loses packets due to its simple ALOHA like MAC, our Q-learning agent may be able to successfully deliver the packet with some delay. Since such a delivery may happen upon multiple retries, it is natural that some packets will experience long delays. Since LoRa applications are not typically time-sensitive, our approach is still a practical choice for LoRa.

5.2.2 Q-Values and Action Selection Approach. The goal of the Q-learning agent is to find the best action given the current state. It tries to find the best policy that maximizes the total future reward by learning from its own interaction with the environment. It quantifies the quality of an action at a particular state through *Q-values.* It represents the currently expected total future reward and is initialized to zero. Let the Q-value associated with action a and state

s be $Q(s,a)$. Through trial and experience, the agent learns how good some action was. The Q-values of the state and action pairs change through learning and finally represent the absolute value. After convergence, taking the actions with the greatest Q-values in each state guarantees taking an optimal decision in a single agent environment. The new Q-value of pair $\{s,a\}$ in state s' after taking action a in state s is computed as the sum of old Q-value and a correction term

$$Q(s', a) = Q(s, a) + \gamma(r(s, a) - Q(s, a)). \tag{1}$$

The learning constant, γ, prevents the Q-values from changing too fast and thus oscillating. The nodes take actions and update the Q-values, repeating until the Q-values no longer change. The agent stores the Q-values in a look-up table (Q-table) of size $|S|.|A|$. The agent chooses its next action in one of the following ways: (1) it chooses the action with the highest Q-value at its current state, which is called *exploiting* the Q-table. (2) it randomly chooses an action, which is called *exploring* the environment.

Always taking the actions with maximum Q-value (greedy policy) may result in finding locally maximal solutions. On the other hand, selecting always randomly implies ignoring prior experience and spending too much energy to learn the complete environment. We adopt an approach by combining and weighing both which is a prominent approach in machine learning [33]. Specifically, we use ϵ-**greedy**: with probability ϵ the agent takes a random action and with probability $(1-\epsilon)$ it takes the best available action, which is known to yield quick and high quality solutions [33]. The value of ϵ is set to a very high value in the beginning to encourage exploration of environment. This interval is called the *random exploration interval.* After sufficient data has been gathered about the environment, the value of ϵ is decreased to exploit the Q-table. The random exploration interval is set by the network manager. Considering the dynamic nature of the massive coexistence environment with multiple embedded agents, the agents always continue learning. Thus, Q-values are updated after every action.

Note that, every node acts as an agent whose Q-table size is $O(|A|.|S|)$, where $|S|$ is the number states and $|A|$ is the number of actions. Thus, the action and state space size is feasible even in memory constrained devices. The Q-table is updated through simple arithmetic operation (Eq. 1), which is also feasible for low-power nodes.

6 EXPERIMENTS

In this section, we evaluate the performance of proposed Q-learning agents under different coexistence scenarios in both indoor and outdoor deployments.

6.1 Implementation

We use the Dragino SX1276 LoRa transceiver HAT [17] on raspberry pi 3 [28], equipped with half-duplex transceivers, as LoRa nodes. Each node uses a custom-built Q-learning agent on top of LMIC 1.6 LoRa/LoRaWAN [16] library. All nodes use 15dBm Tx power and packet size of 10 bytes to emulate a large scale network.

The channel and spreading factor are selected based on the action of the Q-learning agent. We use the RAK2245 HAT [11] on Raspberry Pi 3 as the gateway with a local chirpstack network server [7] for the primary network.

For emulating a large coexisting network, we used two USRP B210 devices operating in conjunction with GNU-radio as coexisting nodes. The coexisting nodes transmitted using ASK modulation on the same channels as the LoRa nodes with a Tx power of 15dBM. We use two different coexistence scenarios: namely, partial channel coexistence and periodic coexistence. In *partial channel coexistence*, the nodes transmit using 3 channels in the US 902-928MHz band and the coexisting nodes create severe interference on two of the available channels by continuously transmitting on them. In *periodic coexistence*, we emulate a massive coexistence scenario with high density traffic by limiting the number of available channels to 1, while enabling the coexisting node to transmit on the same channel with a packet rate of 4 packets per second.

Baseline. To our knowledge, this is the first work considering the coexistence of LoRa networks with many unknown and uncoordinated networks that does not require additional hardware. Thus, we use the traditional LoRaWAN MAC implemented in LMIC 1.6 library (with 3 s retransmission backoff) as the baseline for our approach. When operating on LoRaWAN MAC, the nodes use the Adaptive Data Rate (ADR) algorithm [3] to select the Tx parameters.

For every experiment, we report the PRR, Tx energy consumption per node and the average number of Tx attempts per packet. We consider a packet is successful if it was received at the gateway and the corresponding ACK was received by the node. The energy consumed in communication is a major contributor in the total energy overhead of LPWAN nodes used in large-scale physical deployments [9]. Thus, we report the energy consumed in Tx following the energy model described in [31]. For all experiments, we compare the performance of the Q-learning agent after its random exploration interval with LoRaWAN.

6.2 Indoor Experiments

6.2.1 Experiments with USRP coexisting nodes. We use 11 LoRa nodes and a single gateway in this experiment. Figure 3 shows the locations of the nodes and the gateway in our indoor deployment in an area of 42 ft ×18 ft, where the nodes are marked with dark circles labeled from N1 to N11. The indoor deployment is inside a building in a suburban area. The default parameters for the Q-learning agent are: $\epsilon = 0.1$, learning rate, $\gamma = 0.5$, maximum delay, $D = 10s$, and $\phi = 1s$ For this experiment, the nodes used a fixed packet rate of 60 packets per hour. Note that in LoRaWAN nodes may retransmit each packet up to 8 times. Furthermore, each retransmission procedure can take up to 4-5 seconds due to the two long receive windows after each transmission. The packet rate of 60 packets per hour was chosen to accommodate these retransmissions. All results are based on 100 packets after the random exploration interval.

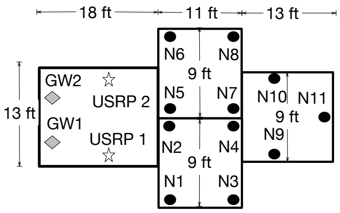


Figure 3: Indoor deployment.

Results under Varying Exploration Interval. The random exploration interval controls the time until the agent takes random actions to explore the state-action space. We run experiments in each coexistence scenario with 5 nodes and varying exploration interval to show its effect on the performance of the Q-learning agent.

Partial channel coexistence. Figure 4 shows the results under partial channel coexistence where the exploration interval varies from 0 to 20 minutes. Although the PRR for Q-learning is slightly lower than LoRaWAN for 0 min exploration interval, energy consumption (Figure 4(b)) and avg Tx attempts (Figure 4(c)) show that Q-Learning improves the performance over time. Nevertheless, as the exploration interval increases, the PRR also increases for Q-learning and the average number of Tx attempts decreases. We see that the energy consumption per node in Figure 4(b) increases when using random exploration interval of 15 minutes. This can be the result of an agent selecting an action with a higher spreading factor. However, we note that the Tx energy consumption is still lower than LoRaWAN while maintaining the same PRR. For our subsequent indoor experiments with partial channel coexistence, we use a 10 minute exploration interval as it is sufficient to gather enough knowledge of the environment, as shown by the PRR, energy consumption and average transmission attempts.

Periodic coexistence. In periodic coexistence, we vary the exploration interval from 10 to 50 minutes and report the results in Figure 5. In Figure 5(a), LoRaWAN PRR drops to 0, as the pure ALOHA MAC is not sophisticated enough to handle such dense coexistence scenario, while Q-learning is able to ensure at least 50% PRR. As the exploration interval increases 10 to 40, the PRR for Q-learning also increases, while at 50 min of exploration interval the PRR slightly drops due to over exploration. A similar result can be observed for avg Tx attempts in Figure 5(c). This indicates that too much exploration can adversely effect the learning performance. In Figure 5(b), Q-learning agent has 7 times less energy consumption on average than LoRaWAN. For our next experiments with periodic coexistence, we use exploration interval of 40 minutes as the results indicate that Q-learning agents are able to acquire sufficient knowledge about the environment during this time.

Results under Varying Number of Nodes. Next, we test the scalability of our approach by varying the number of primary nodes.

Partial channel coexistence. We run an experiment under partial channel coexistence with 10 minute random exploration interval for the Q-learning agent. Figure 6 shows the results for this experiment where the number of primary nodes is varied from 2 to 11. In Figure 6(a), the PRR of both Q-learning and LoRaWAN decreases slightly as the number of nodes increase to 11. Overall, in this experiment, we do not see a significant effect on PRR for coexistence, as only 2 of the available channels are impacted. However, this leads to higher Tx energy consumption and average number of Tx per packet for LoRaWAN as shown in Figures 6(b) and 6(c), respectively. Interestingly, for the 2-node case. the Q-learning agents made slightly higher Tx attempts than LoRaWAN, while consuming less energy. This is due to the Q-learning agents selecting an action with lower spreading factor than LoRaWAN, which resulted in significantly less energy consumption. A small improvement in Tx attempts for Q-learning results in a large improvement in terms of energy consumption for the same reason in the 11 node case.

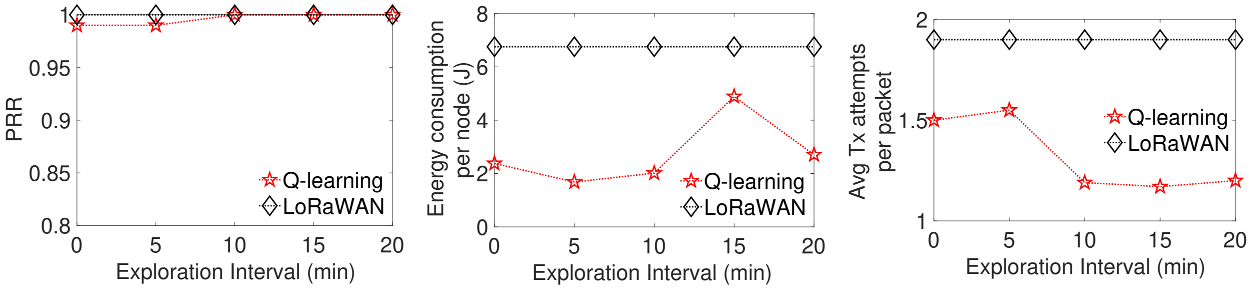
Periodic coexistence. For the periodic coexistence case, we use a random exploration interval of 40 minutes and up to 11 node. Figure 7 shows the results for this experiment. In Figure 7(a), we observe that Q-learning agents are able to maintain at least 99% PRR in all cases. In contrast, all packets from LoRaWAN nodes were interfered and the gateway did not receive any packets as the pure ALOHA MAC used in LoRaWAN is not able to ensure packet reception. In Figure 7(c), the Tx attempt per packet for the Q-learning agent increases slightly as the number of nodes increase, however it does not impact the Tx energy consumption severely as seen in 7(b). Again, this is due to the choice of different SFs across different experiments. Overall, these results show that our embedded Q-learning approach is scalable.

Results under Varying Rate of Coexistence Traffic. We evaluate the efficiency of our approach under varying coexistence traffic. In Figure 8, we use a primary network of 5 nodes with periodic coexistence and vary the packet rate from 2 to 3.6 packets/second. In Figure 8(a), the PRR for LoRaWAN sharply decreases with increasing coexistence packet rate, while the Q-learning agents are able to maintain at least 99% PRR at all cases. We observe tha our approach provides 87% decrease in energy consumption and 68% decrease in Tx attempts on average Figure 8(b) and 8(c), respectively. Overall, our approach can provide significant performance improvement under varying coexistence scenarios.

6.2.2 Experiments under Coexisting LoRa Networks. For this experiment, we use two separate LoRa networks, each with 5 nodes and a single gateway. We use the Dragino LG308 LoRa gateway [1] with The Things Stack network server [2] in the coexisting LoRa network. To emulate a large deployment, we limit the number of available channels to 1, and fix the SF to 10. Furthermore, we use a high packet rate of 6 packets per minute, and to accommodate 6 packets per minute we limit the maximum number of retransmissions to 1. In this experiment, both primary and coexisting networks employed Q-learning agents with maximum delay, $D = 2s$ and $\phi = 200$ ms. Note that as both networks are learning and adapting to each other's transmission patterns, the interference observed at each agent is highly dynamic. Moreover, the coexisting network behaves randomly for the random exploration interval. After the random exploration interval when each agent starts to act based on its Q-table, any random change in the transmission pattern is detrimental to the performance. To avoid such issues, we set an extremely low value of ϵ after the random exploration interval for all agents. Specifically, we used $\epsilon = 10^{-5}$ in this experiment.

Results under varying penalty for missed packets. Under this setup, we first run some experiments to fix the Q-learning parameters. We first vary the penalty for each missed packet and show the PRR, average Tx attempts per packet and transmission energy consumption per node in Figure 9. In Figure 9(a), we see that using a penalty of -0.5, both network have higher PRR compared to other settings. Also, the average transmission attempts and the transmission energy consumption are lower using this setup. Thus, for our next experiments we use a penalty of -0.5.

Results under varying exploration interval. We next vary the random exploration interval from 50 s to 500 s in Figure 10. According to Figure 10(a), as the exploration interval increases, the PRR also increases for both networks and the highest PRR is observed

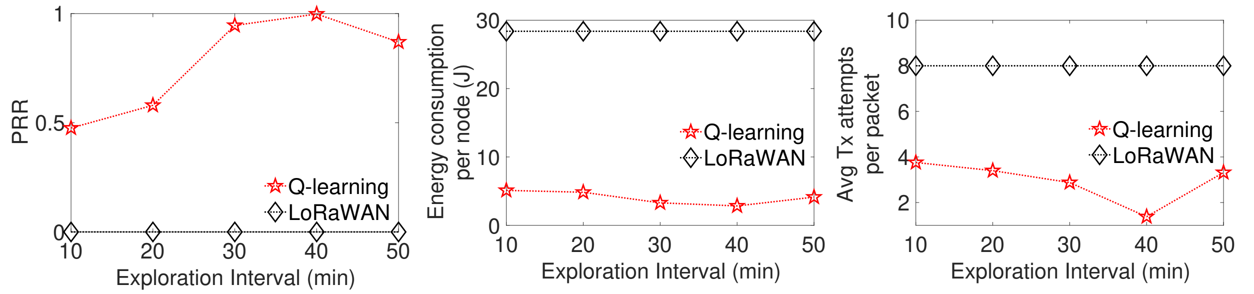


(a) PRR

(b) Tx Energy Consumption

(c) Avg Tx attempts

Figure 4: Results under varying exploration interval for partial channel coexistence.

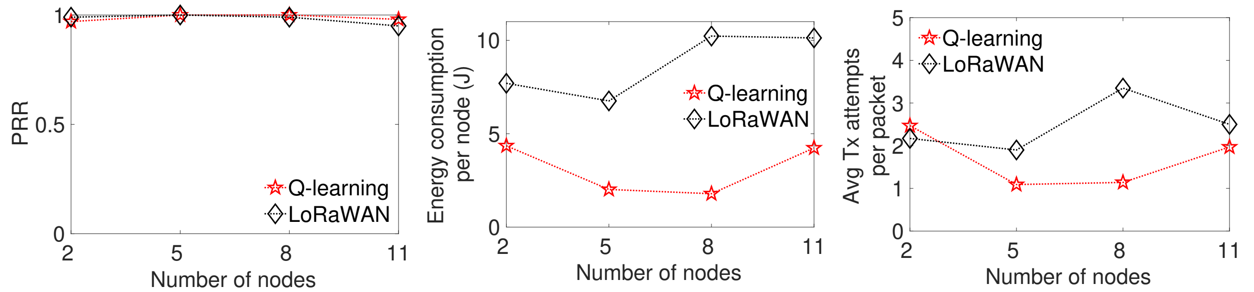


(a) PRR

(b) Tx Energy Consumption

(c) Avg Tx Attempts

Figure 5: Results under varying exploration interval for periodic coexistence.

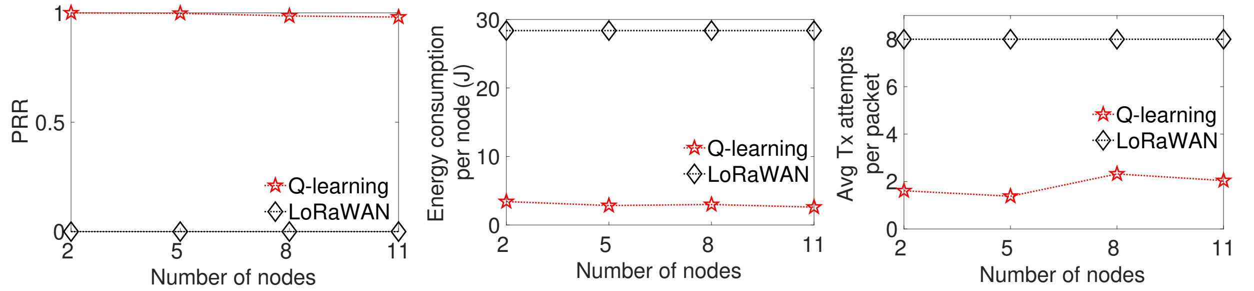


(a) PRR

(b) Tx Energy Consumption

(c) Avg Tx Attempts

Figure 6: Results under varying number of nodes for partial channel coexistence.



(a) PRR

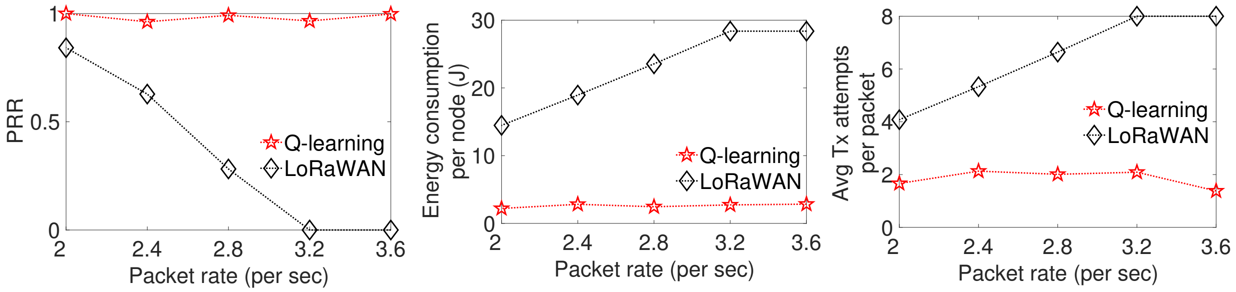
(b) Tx Energy Consumption

(c) Avg Tx Attempts

Figure 7: Results under varying number of nodes for periodic coexistence.

using exploration interval 500 s. Also, the average transmission attempts and energy consumption are the lowest for this exploration interval, as shown in Figure 10(b) and Figure 10(c), respectively.

Comparison with LoRaWAN using one retransmission. Next, we fix the penalty for missing packets to -0.5 and the exploration interval to 500 s and compare our approach with LoRaWAN. The PRR, average transmission attempts, and energy consumption for each node in the network are shown in Figure 11. We see in Figure 11(a) the nodes are able to improve their PRR using Q-learning.

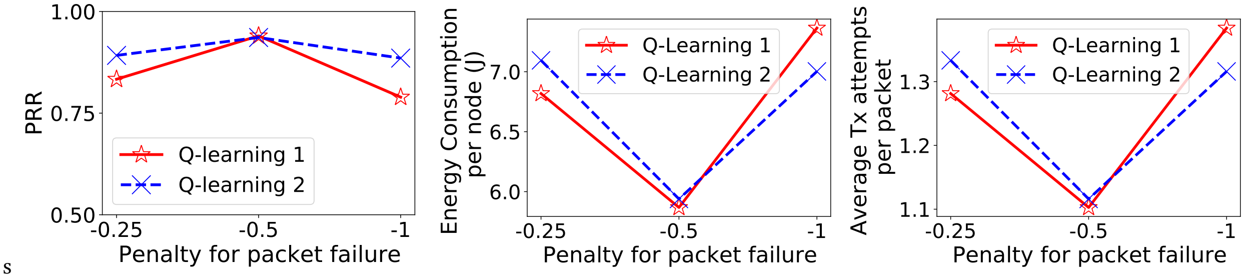


(a) PRR

(b) Tx Energy Consumption

(c) Avg Tx Attempts

Figure 8: Results under varying coexistence node traffic.

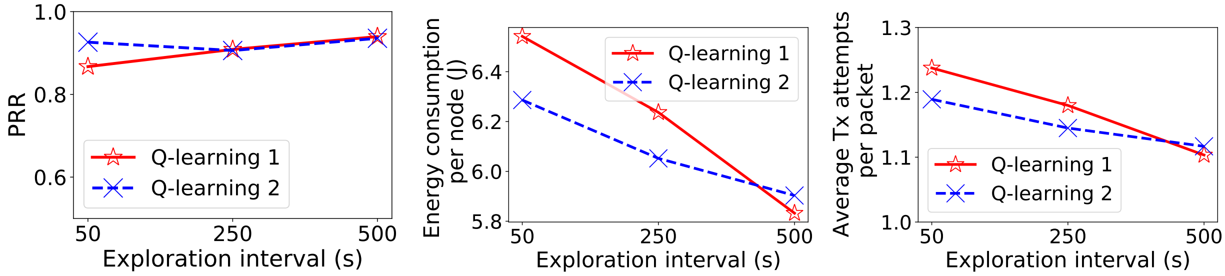


(a) PRR

(b) Tx Energy Consumption

(c) Avg Tx Attempts

Figure 9: Results under varying penalty for missed packet.



(a) PRR

(b) Tx Energy Consumption

(c) Avg Tx Attempts

Figure 10: Results under varying exploration interval.

Specifically, the average PRR for both networks is 94% using Q-learning, compared to 84% and 80% using LoRaWAN in primary and coexisting network, respectively. Furthermore, the nodes are able to reduce the transmission attempts by 33% on average in the coexisting network (Figure 11(b)) and similarly improving the transmission energy consumption by 30% (Figure 11(c)).

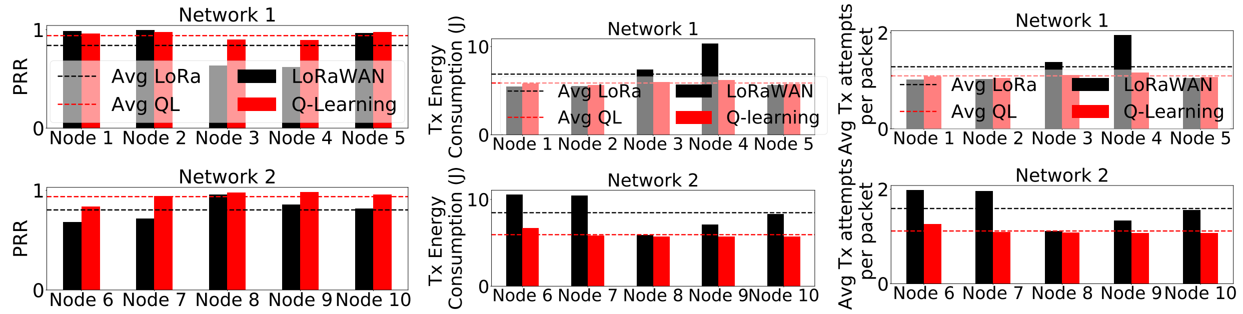
Comparison with LoRaWAN using no retransmission. To test our approach under harsher scenarios, we disable retransmissions completely for all nodes. Thus, in this experiment, each node is making exactly one transmission attempt per packet. As the average transmission attempt and transmission energy consumption does not change in this setup, we show comparison in PRR only. Under this setup, we again vary the negative reward from -0.25 to -1 and report the results in Figure 12. We observe that even without retransmission, Q-learning agents are able to maintain PRR>85% for most cases while the best performance is observed using a negative reward of -1.

Next, we vary the exploration interval and observe that an exploration interval of 250 s performs the best in terms of PRR in Figure 13. The agent requires some time to gather enough knowledge about the environment in this case, thus the performance is lower with low exploration interval.

Finally, in Figure 14, we compare our approach with LoRaWAN. We observe that the average PRR improves by 77% in coexisting network and 65% in the primary network. Furthermore, we observe that using LoRaWAN some nodes were not able to deliver any packets to the gateway (node 3 and node 7) as all nodes immediately transmit their packet in default LoRaWAN and subsequently face collisions. There is a fair distribution of PRR across nodes for Q-learning due to the intelligent actions taken by the agents.

6.3 Outdoor Experiments

Figure 15 shows the distance and location of the gateway and nodes at the Wayne State University campus in Detroit, Michigan. The gateway and the coexisting nodes are located inside a stationary car and the primary nodes are placed on the locations shown on the map. We use up to 4 primary nodes, 2 USRP coexisting nodes and



(a) PRR (b) Tx Energy Consumption (c) Avg Tx Attempts

Figure 11: Results under coexistence from LoRa network.

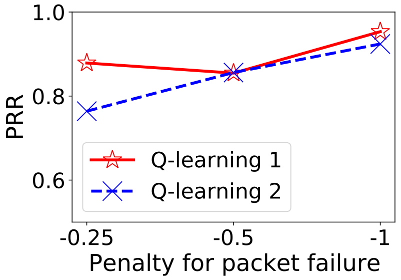


Figure 12: Results under varying penalty for missed packet

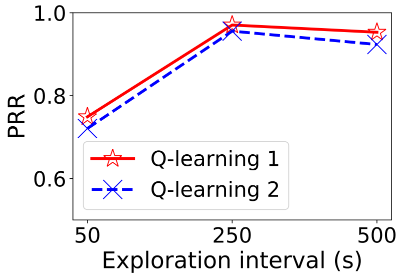


Figure 13: Results under varying exploration interval

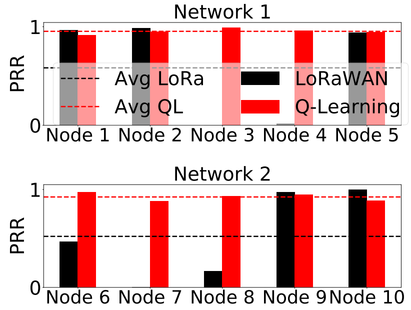


Figure 14: PRR comparison with LoRaWAN under no retransmissions

one gateway in this experiment. Due to the lack of outside deployment, only USRP coexisting nodes were used. For the outdoor experiments, we use a random exploration interval of 5 minutes. All results are collected for 25 packets after the random exploration interval.

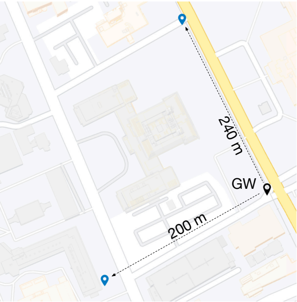


Figure 15: **Outdoor deployment.**

Results under Varying Number of Nodes. Figure 16 shows the performance of Q-learning agents compared to LoRaWAN under partial channel coexistence with varying number of nodes. In Figure 16(a) as the number of nodes increases the PRR for both LoRaWAN and Q-learning decreases. However, Q-learning agents provide better PRR than LoRaWAN in all cases. When the primary network consists of 3 nodes, Q-learning agents provide 70% PRR, compared to 47% in LoRaWAN. In Figure 16(b), Q-learning agents consume 65% less Tx energy on average than LoRaWAN. In Figure 16(c), Tx attempts for both approach increase as the number of nodes increase. However, Q-learning agents incur less Tx attempts than LoRaWAN. In this result, the performance for both approaches is lower than the result in indoor deployment. Due to the urban location, the channels had a significantly higher level of residual noise than the indoor deployment, which was at a suburban location. Furthermore, the gateway antenna used in our experiment is not sophisticated enough to handle such noisy environments. In Table 4, we observe that the average SNR and received signal strength indicator (RSSI) for 100 packets received at the gateway for the outdoor deployment are significantly lower than that for the indoor deployment. This led to overall lower PRR and higher Tx attempts than the indoor deployment. Due to the elevated levels of noise, the Q-learning agents have to continuously re-learn the environment, leading to a lower performance at higher number of nodes.

7 SIMULATION

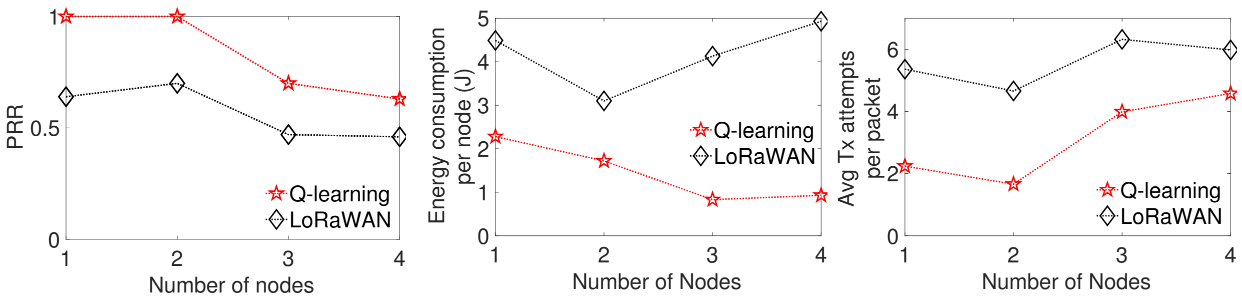
	indoor	outdoor
Avg SNR	6.3	-2.17
Avg RSSI	-68	-79

Table 4: **Avg SNR and RSSI**

To complement the experimental results, we conducted large scale simulation in NS-3 [26] using a single gateway, up to 500 coexisting nodes, and 400 primary nodes. In all simulations, the nodes were randomly located on a disc of radius 6Km.

7.0.1 *Simulation Setup.*

We use a custom-built Q-learning agent which governed the MAC protocol for each node following our approach on top of the LoRaWAN NS3 module in [21]. All nodes and the gateway use 8 channels in the US 915MHz band. The coexisting LoRa nodes used 8 retransmission per packet to create severe coexistence. In all simulations, we use $\epsilon = 0.1$, learning rate, $\gamma = 0.5$, maximum delay, $D = 10s$, and $\phi = 1s$. We vary the number of primary and coexisting nodes and compare the performance of the Q-learning agent after the random exploration interval with LoRaWAN. The metrics used for simulations is the same as in physical experiments.



(a) PRR

(b) Tx Energy Consumption

(c) Avg Tx Attempts

Figure 16: Results under varying number of nodes for partial channel coexistence (outdoor).

7.1 Performance of Q-learning agent under varying number of primary nodes

In this simulation, we evaluate our approach by varying the number of primary nodes from 100 to 400. For each run, the number of coexisting nodes was equal to the number of primary nodes. The minimum interarrival time for the primary nodes is set to $100 + 10x$ seconds, where x is an integer chosen randomly in the range $[1, 10]$, while for the coexisting nodes it is set to an integer chosen randomly from the range $[50, 150]$ s. The random exploration interval and the simulation time is set to 5 hours and 10 hours, respectively. Under this setup, we vary the number of primary nodes and show the results in Figure 17.

As the number of primary nodes increases, the PRR for both approaches decrease in Figure 17(a). However, the PRR for Q-learning agents is better than LoRaWAN for all cases. For the same simulation, we observe that the average Tx attempts per packet and energy consumption per node are increasing as the number of primary nodes increases (Figure 17(b) and 17(c)). Again, the Q-learning agents are able to maintain a lower energy consumption and average Tx attempts than LoRaWAN throughout the simulation. At 400 nodes, the Tx attempt per packet for Q-learning agents and LoRaWAN are similar in Figure 17(c), however, as seen in Figure 17(a) and 17(b), the Q-learning agents improve the PRR and Tx energy consumption for the same setup. On average, the Q-learning agents provide 28% improvement in PRR and 38% improvement in Tx energy consumption per packet. Overall, these result indicate that the Q-learning agents are scalable to a large number of nodes.

7.2 Performance under Varying Number of Coexisting Nodes

For this simulation, we used 100 primary nodes and varied the number of coexisting nodes from 100 to 500. The minimum packet inter-arrival time for the primary nodes was set to $50 + 10x$ seconds, where x is an integer chosen randomly in the range $[1, 10]$, while for the coexisting nodes it is set as an integer chosen randomly from the range [50,150] s. The simulation time was set to 30 Hours and the random exploration interval was set to 15 hours. Under this setup, we vary the number of coexisting nodes and show the results for Q-learning and LoRaWAN in Figure 18.

In Figure 18(a), we see the PRR of both LoRaWAN and Q-learning decreases as the number of coexisting nodes increases. However, the Q-learning agents perform significantly better in all cases. On average, Q-learning agents are able to provide a PRR of 77%, while for LoRaWAN the average PRR is 49%. In Figure 18(b) we see that the Tx energy consumption for LoRaWAN increases with the number of coexisting nodes, but for Q-learning it remains stable in all cases. A similar trend is observed in Figure 18(c), where the average retransmission attempts per packet for Q-learning remains unaffected with increasing number of coexisting nodes, but for LoRaWAN it increases. Q-learning is able to reduce the energy consumed in transmission by 47% on average over LoRaWAN while maintaining a better PRR throughout the simulation. Thus, the Q-learning agent significantly outperforms LoRaWAN in large-scale networks.

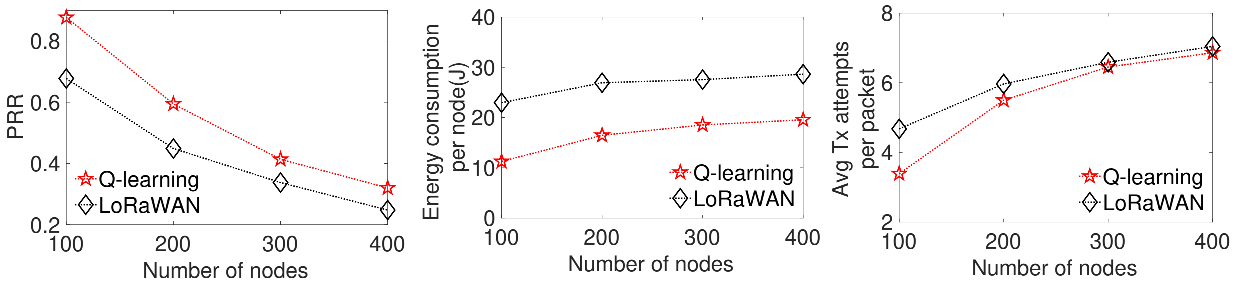
7.3 Performance under Dynamic Coexistence Traffic

In this simulation, we evaluate our approach under system dynamics. We fix the number of primary and coexisting nodes to 100. Initially, the transmissions intervals for the coexisting nodes were chosen randomly from the range $[50, 150]$ s. However, after the random exploration interval (5 hours), we change the transmission interval of some of the coexisting nodes to an integer chosen randomly from the range $[10, 50]$ s. We vary the number of coexisting nodes with dynamics from 20 too 100 and show the results in Figure 19.

In Figure 19(a) and 19(b), we notice that as the number of coexisting nodes with dynamics increase, the PRR decreases, while the energy consumption and average transmission attempts per packet increases for both approaches. However, in all cases, Q-learning outperforms regular LoRa (21% improvement in PRR on avg). This shows that even under dynamic coexistence traffic, our approach can converge to a feasible solution. As the Q-learning agents never stop updating the Q-values, they are able to learn the dynamic coexisting traffic pattern. Overall, this result shows that our approach is applicable for dynamic coexistence scenarios.

8 CONCLUSION

To improve the performance of a LoRa network under coexistence with many independent networks, we have proposed the design of a novel *embedded learning agent* based on Q-learning at LoRa nodes. The agent exploits transmission acknowledgments as feedback from the network based on what a node makes transmission decisions. To the best of our knowledge, this is the *first* Q-learning approach for handling coexistence for any low-power network. We have evaluated our approach through experiments indoors and outdoors

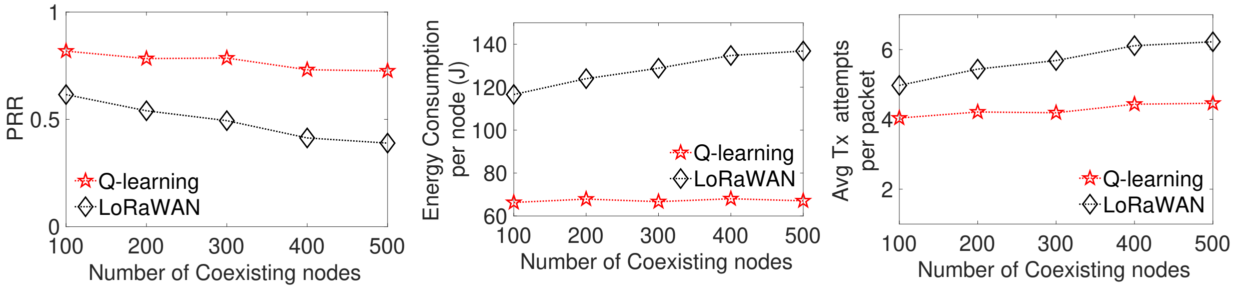


(a) PRR

(b) TX Energy Consumption

(c) Avg Tx Attempts

Figure 17: Results under varying number of primary nodes.

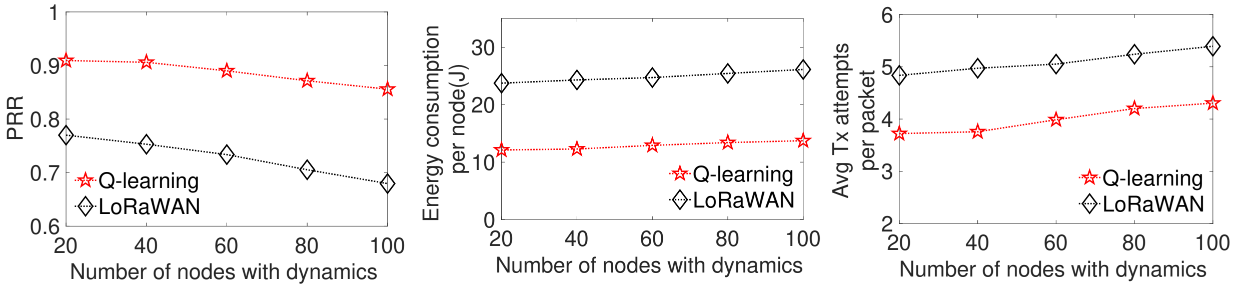


(a) PRR

(b) Tx Energy Consumption

(c) Avg Tx Attempts

Figure 18: Results under varying number of coexisting nodes.



(a) PRR

(b) TX Energy Consumption

(c) Avg Tx Attempts

Figure 19: Results under varying number of primary nodes with dynamic coexistence traffic.

under various coexistence scenarios. The results show that our Q-learning approach on average achieves an improvement of 46% in packet reception rate while consuming 66% less energy compared to LoRaWAN in outdoor deployments. In indoor experiments, we have observed some coexistence scenarios where all the packets are lost under LoRaWAN while our approach enables 99% packet reception rate with up to 90% improvement in energy consumption.

ACKNOWLEDGMENTS

The authors would like to thank Dali Ismail for providing his assistance during Outdoor experiments. The work was supported by NSF through grants CNS-2301757, CNS-2211640, CAREER-2306486, CCF-2118202, CNS-2306745, by ONR through grant N00014-23-1-2151, and by the UNLV Troesh Center.

REFERENCES

[1] [n.d.]. LG308 Gateway. https://www.dragino.com/products/lora-lorawan-gateway/item/140-lg308.html.

[2] [n.d.]. The Things Stack. https://www.thethingsindustries.com/stack/.

[3] LoRa alliance. 2017. LoRaWAN specification. https://lora-alliance.org/resource-hub/lorawanr-specification-v11.

[4] R. Arroyo-Valles, R. Alaiz-Rodriguez, A. Guerrero-Curieses, and J. Cid-Sueiro. [n.d.]. Q-Probabilistic Routing in Wireless Sensor Networks. In *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*.

[5] Orfanidis Charalampos, Feeney Laura Marie, and Gunningberg Per. [n.d.]. Investigating interference between LoRa and IEEE 802.15. 4g networks. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*.

[6] Gonglong Chen, Wei Dong, and Jiamei Lv. [n.d.]. Lofi: Enabling 2.4 ghz lora and wifi coexistence by detecting extremely weak signals. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*.

[7] cstack 2020. ChirpStack LoRaWAN Network Server. https://www.chirpstack.io

[8] Sezana Fahmida, Venkata Prashant Modekurthy, Dali Ismail, Aakriti Jain, and Abusayeed Saifullah. 2022. Real-Time Communication over LoRa Networks. In *2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 14–27.

[9] Sezana Fahmida, Venkata P Modekurthy, Mahbubur Rahman, Abusayeed Saifullah, and Marco Brocanelli. 2020. Long-lived LoRa: Prolonging the lifetime of a LoRa network. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*. IEEE, 1–12.

[10] A. Galindo-Serrano and L. Giupponi. 2010. Distributed Q-Learning for Aggregated Interference Control in Cognitive Radio Networks. *IEEE Transactions on Vehicular Technology* (2010).

[11] gatewayhat 2020. RAK2245 HAT/Pilot Gateway. https://cdn-shop.adafruit.com/product-files/4284/4284_Get_Start_with_RAK2245_Pi_HAT_V2.4R.pdf

[12] O. Georgiou and U. Raza. 2017. Low Power Wide Area Network Analysis: Can LoRa Scale? *IEEE Wireless Communications Letters* (2017).

[13] Xu Huang, Jie Jiang, Shuang-Hua Yang, and Yulong Ding. 2020. A Reinforcement Learning Based Medium Access Control Method for LoRa Networks. In *2020 IEEE International Conference on Networking, Sensing and Control (ICNSC)*.

[14] L. Krupka, L. Vojtech, and M. Neruda. 2016. The issue of LPWAN technology coexistence in IoT environment. In *2016 17th International Conference on Mechatronics - Mechatronika (ME)*.

[15] Zhenzhen Liu and I. Elhanany. [n.d.]. RL-MAC: A QoS-Aware Reinforcement Learning based MAC Protocol for Wireless Sensor Networks. In *2006 IEEE International Conference on Networking, Sensing and Control*.

[16] lmic 2020. LMIC 1.6 for RPI GPS/LoRa HAT. https://github.com/wklenk/lmic-rpi-lora-gps-hat

[17] lorahat 2020. Dragino GPS/LoRa HAT. https://www.dragino.com/products/lora/item/106-lora-gps-hat.html

[18] LoRaleader2 [n.d.]. https://www.i-scoop.eu/internet-of-things-guide/iot-network-lora-lorawan/.

[19] LoRaWAN [n.d.]. LoRaWAN. https://www.lora-alliance.org.

[20] Ziyang Lu, Chen Zhong, and M Cenk Gursoy. [n.d.]. Dynamic channel access and power control in wireless interference networks via multi-agent deep reinforcement learning. *IEEE Transactions on Vehicular Technology* ([n. d.]).

[21] Davide Magrin, Marco Centenaro, and Lorenzo Vangelista. [n.d.]. Performance evaluation of LoRa networks in a smart city scenario. In *Communications (ICC), 2017 IEEE International Conference On*.

[22] S Marek. 2016. Comcast Will Test LoRaWAN IoT Networks in Two Markets. https://www.sdxcentral.com/articles/news/comcast-will-test-lora-iot-network-two-markets/2016/10/.

[23] Kais Mekkia, Eddy Bajica, Frederic Chaxela, and Fernand Meyer. 2018. A comparative study of LPWAN technologies for large-scale IoT deployment. In *ICT Express*.

[24] Florian Meyer and Volker Turau. [n.d.]. QMA: A Resource-efficient, Q-learning-based Multiple Access Scheme for the IIoT. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*.

[25] Konstantin Mikhaylov, Juha Petäjäjärvi, and Janne Janhunen. [n.d.]. On LoRaWAN scalability: Empirical evaluation of susceptibility to inter-network interference. In *2017 European Conference on Networks and Communications (EuCNC)*.

[26] ns3 [n.d.]. NS-3. https://www.nsnam.org/.

[27] Z. Qin, Y. Liu, G. Y. Li, and J. A. McCann. 2017. Modelling and analysis of low-power wide-area networks. In *2017 IEEE International Conference on Communications (ICC)*.

[28] rpi3 2020. Raspberry Pi 3 B+. https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

[29] Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Jie Liu, and Ranveer Chandra. 2018. Low-Power Wide-Area Networks over White Spaces. *ACM/IEEE Transactions on Networking* (2018).

[30] semtech [n.d.]. Semtech Sensor Innovation Forum. https://www.semtech.com/company/press/semtech-to-keynote-flexible-easy-to-deploy-iot-solutions-at-the-iot-and-sensor-innovation-forum.

[31] Semtech. 2019. SX1276 Datasheet. https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001OKx/JUYM3TvBMenQzU4LS8ZJcM58BIjCcoZcUpHV0gnZ.y0

[32] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. [n.d.]. Concurrent Interference Cancellation: Decoding Multi-Packet Collisions in LoRa. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*.

[33] R. S. Sutton and A. G Barto. 1998. *Reinforcement Learning: An Introduction.* The MIT Press.

[34] Andrew Tanenbaum and David Wetherall. 2011. *Computer Networks (5th Edition)*. Prentice Hall. Chapter 4.

[35] Shuai Tong, Zhenqiang Xu, and Jiliang Wang. 2020. CoLoRa: Enabling Multi-Packet Reception in LoRa. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. IEEE Press, 2303–2311. https://doi.org/10.1109/INFOCOM41043.2020.9155509

[36] usrp [n.d.]. http://www.ettus.com/product/details/UB210-KIT.

[37] Thiemo Voigt, Martin Bor, Utz Roedig, and Juan Alonso. [n.d.]. Mitigating Inter-network Interference in LoRa Networks. In *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*.

[38] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. [n.d.]. mLoRa: A Multi-Packet Reception Protocol in LoRa networks. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*.

[39] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2020. FTrack: Parallel Decoding for LoRa Transmissions. *IEEE/ACM Transactions on Networking* (2020).

[40] Zhenqiang Xu, Pengjin Xie, and Jiliang Wang. 2021. Pyramid: Real-Time LoRa Collision Decoding with Peak Tracking. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*.

[41] D. Yang, Y. Xu, and M. Gidlund. 2010. Coexistence of IEEE802.15.4 based networks: A survey. In *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*.

[42] Dong Yang, Youzhi Xu, and Mikael Gidlund. 2011. Wireless Coexistence between IEEE 802.11- and IEEE 802.15.4-Based Networks: A Survey. *International Journal of Distributed Sensor Networks* (2011).

[43] K. L. A. Yau, P. Komisarczuk, and D. T. Paul. 2010. Enhancing network performance in Distributed Cognitive Radio Networks using single-agent and multi-agent Reinforcement Learning. In *IEEE Local Computer Network Conference*.

[44] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. [n.d.]. Review: Reinforcement Learning for Context Awareness and Intelligence in Wireless Networks: Review, New Features and Open Issues. *J. Netw. Comput. Appl.* ([n. d.]).

[45] F. R. Yu, V. W. S. Wong, and V. C. M. Leung. 2008. A New QoS Provisioning Method for Adaptive Multimedia in Wireless Networks. *IEEE Transactions on Vehicular Technology* (2008).

[46] Yi Yu, Lina Mroueh, Shuo Li, and Michel Terré. [n.d.]. Multi-Agent Q-Learning Algorithm for Dynamic Power and Rate Allocation in LoRa Networks. In *2020 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*.

[47] Hailu Zhang, Minghui Min, Liang Xiao, Sicong Liu, Peng Cheng, and Mugen Peng. [n.d.]. Reinforcement learning-based interference control for ultra-dense small cells. In *2018 IEEE Global Communications Conference (GLOBECOM)*.