POLITECNICO DI TORINO

M.Sc. in Communications and Computer Networks

M.Sc. Thesis

Multi-Agent Deep Reinforcement Learning Power Control in Low Power Wide Area Networks

Supervisors

Prof. Ladislau MATEKOVITS

Candidate

Behnam SOBHANI NADRI

December 2023

Summary

The Internet of Things (IoT) has been widely deployed in Smart Cities, Healthcare, Agriculture, enabling millions of sensors to exchange information with a cloud server. There are some cases in which IoT contributes to monitoring diseases such as diabete which transmission must be error-free with minimum amount of energy consumption. A multi-application reliable communication framework, deployable in medical scenarios is Low Power Wide Area Networks (LPWAN). These networks are rapidly growing, and the future of IoT will mainly rely on long range communication and energy-efficient sensors. The innovative Long Range Wide Area Network (LoRaWAN) technology, leveraging LoRa modulation, has introduced an energy-efficient and reliable communication framework with a large number of IoT sensors. LoRa has been exclusively designed to work in the unlicensed spectra and provide robust communication against noise and external carriers. LoRaWAN, the Medium Access Control (MAC) of this infrastructure, has defined initiative regulations for the nodes to communicate over the same channel. Additionally, it adapts the quality of the communication such as Data Rate and the Spreading Factor (SF) to the channel conditions. Adaptive Data Rate (ADR) mechanism, increases the data rate when the channel condition is good and decreases it to ensure a sustainable communication. Based on the link budget, ADR decides whether the DR should be changed. Controlling the SF helps to achieve a better energy efficiency and longer life cycle in the end-node. However, when the number of the nodes increases in a certain area, ADR is not efficient anymore. ADR decides the transmission power based on the Signal-to-Noise (SNR) ratio from the recent successful transmissions. As this number increases, interference happens for the neighboring nodes sharing the same Data Rate (DR) and SP. Alternatively, the power control algorithm can be developed in such a way to also consider Signal-to-interference-plus-Noise Ratio (SINR). The proposed approach relies mainly on the principles of ADR but implements a more general solution for the problem that we call Dynamic Power Control (DPC) in LPWANs and this study focuses on the LoRaWAN as a flagship of LPWANs.

Acknowledgements

In the name of the Lord of both wisdom and mind.

I am grateful for all love and kindness that I have received during this work. I appreciate my precious supervisor, Dr. Ladislau Matekovits who has been supporting me since the beginning of this work and gave me courage to choose my own path. I would also thank all my friends and colleagues who helped me even a tiny portion of this work. And overall, I would dedicate all my endeavour to my lovely family and I am grateful to have their continuous support during all stages of personal my life. I wish all those happiness and love in their life.

"می نوش دمی که عمر جاودانی این است
خود حاصلت از دور جوانی این است
هنگام گل و باده و یاران سرمست
خوش باش دمی که زندگانی این است"
Omar Khayyam

Table of Contents

List of Tables

List of Figures

Acronyms

MARL
Multi-Agent Deep Reinforcement Learning

CSS
Chirped Sequence Spectrum

LPWAN
Low Power Wide Are Networks

LoRaWAN
Long Range Wide Area Networks

CSI
Channel State Information

DRL
Deep Reinfocement Learning

DNN
Deep Neural Networks

DQL
Deep Q-Learning

DQN
Deep Q-Network

Chapter 1

Introduction

In wireless networks, power control is essential to enhance the quality and efficiency of the system [1]. Control policies are usually defined in the access protocols and supervised in a centralized controller or locally in the end devices (EDs). In this thesis we are specifically interested in power control approaches in energy-constrained devices such as Internet of Things (IoT) sensors. There are several standards that was defined for these type of devices that are low-powered and can transmit in wide areas. The networks with these characteristics are called Low Power Wide Are Networks (LPWAN). They are designed to provide efficient transmission of low amount of data with long range communication. These properties allow them having lower Quality of Service (QoS) but longer life cycle.

1.1 Power Control

Transmit Power Control or Power Allocation is a strategy in wireless networks to dynamically adjust the transmission power in order to minimize the energy consumption and reduce interference. This strategy is deployed in the Medium Access Control (MAC) and is implemented in the Network Card Interfaces (NICs). MAC is an interface between the application layer and the PHY of the device and defines protocols based on the properties and characteristics of the network. For example in IEEE 802.1x standard which is known as WiFi, MAC layer implements the Carrier Sesnse Multiple Access (CSMA) scheme to prevent collisions and enhance the network efficiency. Power control is also defined in the MAC layer of the wireless networks to optimize the transmissions in a communication system.

Transmitters must balance their transmit power to minimize the interference with other nodes. Power Control is an naturally an optimization problem. If a single node increases its power, other transmissions are also affected by the

interference. Thus, we should look at the whole system when we are solving the power control for all of the nodes. We will look at some approaches that were propsed for certain application in wireless networks. Although they are different solution and algorithms, the objective of the power control is almost similar in all networks. Based on the history, power control is categorized as centralized and distributed. In this section We discuss the the general solutions and in the next chapter we will lool at some specific approaches in the LPWANs.

1.1.1 Solutions

1.1.2 Centralized PC

There are cases in which the lack of a controlling strategy of the TX power doesn't necessarily have an impact on the simultaneous transmissions. In Time Division Multiple Access (TDMA) where each node must transmit on a time slot basis, the transceivers synchronize their transmissions with the predefined time slots and the chance of collisions is rare. However, choosing the adequate TX power improves a better energy efficiency in such devices. In these networks, nodes periodically inform the base stations of a channel matrix through the channel state information (CSI). The power is centrally controlled and the impact of interference is minimized in such networks. The main concern in this approach is the amount of bandwidth occupied by these channels when the number of nodes are relatively large. Besides, this strategy is not effective for LPWAN nodes which are designed to work with minimal bandwidth and power consumption.

1.1.3 Distributed PC

Another proposal was distributed power control by the help of network data. Goodman *et al.*[1] proposes a distributed power control algorithm for the cellular data networks where UMTS was the state-of-the-art standard. The standard was based on CDMA and the access to the nodes were given by sharing the same spectrum for the mobile terminals but their signals were spreaded with orthogonal codes. In spreading spectrum techniques terminals can transmit simultaneously on a ALOHA basis. Although the spreading codes are orthogonal to each other, there was some level of correlation among the codes. This correlation creates interference between the nodes when the transmissions are captured by a stronger signal closer to the receiver and it is called capture effect [2].

The objective of the distributed power control is to maximize a utility function for each transmitter. This utility is the number of bits that can be transmitted for each joule of energy. In other words, the ratio of the throughput and the transmit power. It can be proved that this utility function can be maximized for all nodes if

they all struggle to reach the same SINR level at the receiver γ^{opt}. Balancing the SINR of the nodes signals at the receiver, the utility of the whole system can be maximized if and only if they all choose their transmit power to reach γ^{opt}. This is very similar to Non-cooperative game theory where nodes struggle to maximize their utility function and the action of one node affects other nodes.

1.2 Low Power Wide Area Networks: LPWAN

LPWAN is a group of IoT standards with the properties such as large coverage, low transmission data rates with small packet sizes and long battery life devices[3]. The QoS requirements are defined in such a way that there is no obligation for a low-latency and high bandwidth communication. Therefore, it is possible to design an IoT standard which enables low power devices that can last for years.

It is possible to define LPWAN as those type of networks that provide low cost and efficient connectivity in a large area with numerous devices. LPWAN provides a robust connectivity in a large area of a city or a residential suburb.

1.3 Case Study: LoRaWAN

Long Range Wide Area Network, LoRaWAN, is a wireless communication protocol built upon the LoRa (Long Range) technology, designed for enabling efficient communication among low-power devices over long distances. LoRaWAN operates in unlicensed radio frequency bands, allowing for cost-effective and long-range communication, making it suitable for Internet of Things (IoT) applications. It employs a star-of-stars network topology, where end-devices communicate with gateways that forward data to a centralized network server. LoRaWAN offers impressive range capabilities, low power consumption, and supports secure bidirectional communication, making it an ideal solution for various IoT deployments like smart cities, agriculture, industrial monitoring, and more.

1.3.1 LoRa Modulation

The LoRa (Long Range) physical layer is the foundational element of the LoRa technology, crucial in facilitating long-range wireless communication for IoT devices. Operating in the sub-GHz ISM bands, LoRa utilizes a unique modulation technique called Chirp Spread Spectrum (CSS). This modulation method enables LoRa devices to achieve exceptional range capabilities while consuming minimal power. LoRa's CSS modulation employs chirp signals with varying frequencies over time, allowing it to be highly resilient to interference and capable of penetrating obstacles and dense

urban environments. By leveraging different spreading factors and bandwidths, LoRa devices can adapt to various environmental conditions, enabling flexibility in trade-offs between data rate, range, and power consumption. This robust physical layer forms the backbone of LoRa technology, providing the foundation for long-distance, low-power IoT communication in diverse applications.

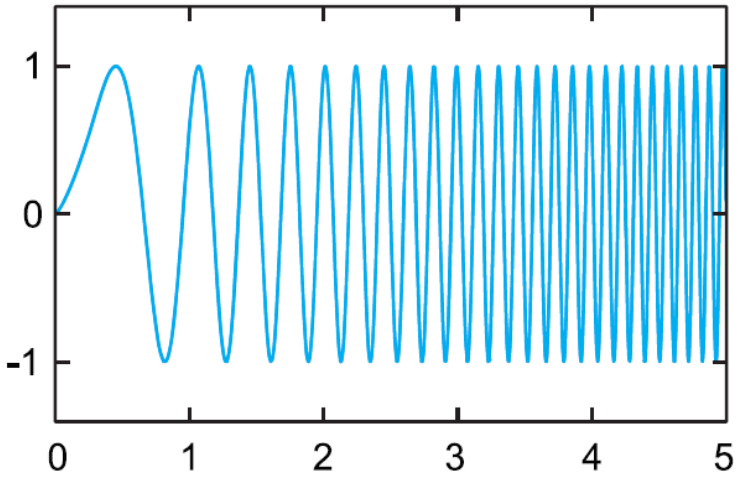

Figure 1.1: Chirped Sequence Spectrum (CSS)

The modulation is patented by LoRa Alliance and uses unlicensed ISM bands as the main radio frequency to transmit signals.

The CDMA approach used in LoRa, to get access to the network, has a probability of packet loss and interference. Spreading codes are orthogonal to each other though when two signals overlap, they have a slight impact on each other and if the receiver is able to compensate the impact can demodulate the signal successfully.

The design of a CSS transceiver is such that, transmitters which share the same SF have a slight chance of interference and overlap of signal on each other (Cite) (Table) shows how signals using different SFs can impact on each other. Co-SF interference is the main problem in CSS that decreases the system performance and efficiency. Overcoming this issue, many algorithms have been proposed based on the capacity, characteristics of the PHY and .. . In the next section, some of the most important power control algorithms will be discussed and more importantly, the ADR approach which is the main PC algorithm in LoRaWAN framework is analyzed on different scenarios and situations.

1.3.2 Network Architecture

The architecture is structured to enable efficient communication between low-power devices and the network infrastructure. It operates on a star-of-stars topology comprising end-devices, gateways, a network server, and an application server. End-devices, often sensors or actuators, transmit data to nearby LoRaWAN gateways known also as concentrators. These gateways receive the signals and forward the data packets to a centralized network server. The network server manages the gateways, ensures optimal data routing, and handles security measures like encryption and authentication. It also manages the MAC layer and handles device join requests. Finally, the application server connects to the network server, processing and storing the received data, and provides an interface for applications and end-users to access and utilize the information. LoRaWAN's architecture is designed for scalability, allowing it to efficiently accommodate a wide range of IoT devices across diverse use cases while maintaining a secure and reliable communication infrastructure.

LoRaWAN node are connected to a couple of concentrators using star topology. Concentrators receive signals from end-nods and forward them to a server via a backhaul network.

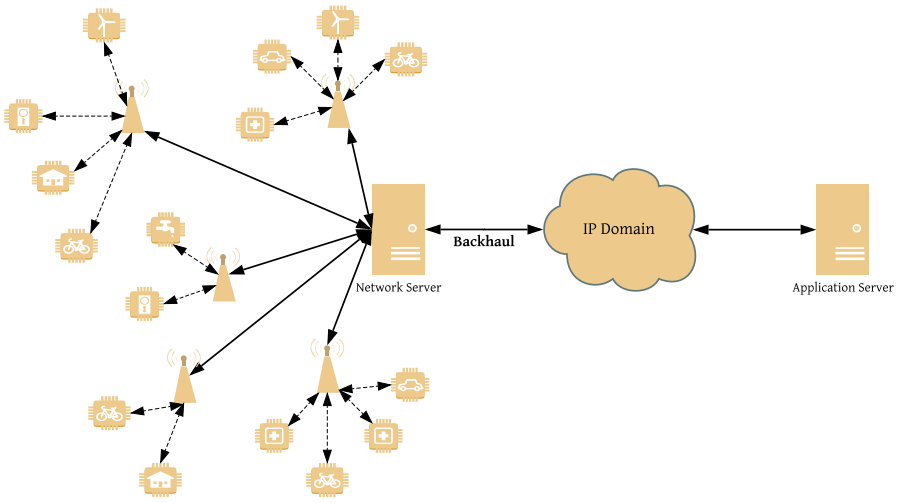

Figure 1.2: An example of an IoT Network based on LPWAN

1.3.3 Device Classes

LoRaWAN classifies devices into three main classes: Class A, Class B, and Class C, each with specific communication characteristics catering to different IoT application requirements.

- **Class A**: are the most common and energy-efficient. They operate on a "Listen Before Talk" principle, meaning they open receive windows for a short duration after transmitting data. After sending data, Class A devices have two short receive windows (RX1 and RX2) scheduled. These windows allow for potential responses from the network or servers. However, after these windows close, the devices go into a longer sleep mode to conserve energy until the next scheduled transmission.

- **Class B**: have additional periodic receive windows apart from the RX1 and RX2 windows present in Class A. These devices synchronize with the network and have scheduled receive slots called ping slots. These slots allow for improved downlink communication synchronization between the end-device and the network, enabling more precise timing for receiving data from the network.

- **Class C:** offer continuous receive capability, unlike Class A and Class B devices that have limited receive windows. Class C devices keep their receive windows open all the time, except when they are transmitting data. This constant listening ability enables almost immediate downlink messages from the network, providing higher responsiveness but consuming more power compared to Class A and Class B devices.

Each class of device serves different IoT use cases, with Class A being the most energy-efficient but having limited downlink communication timing, Class B offering improved downlink timing through scheduled slots, and Class C providing almost continuous receive capability but consuming more power. The choice of class depends on the specific requirements of the IoT application, balancing factors like power consumption, latency, and responsiveness.

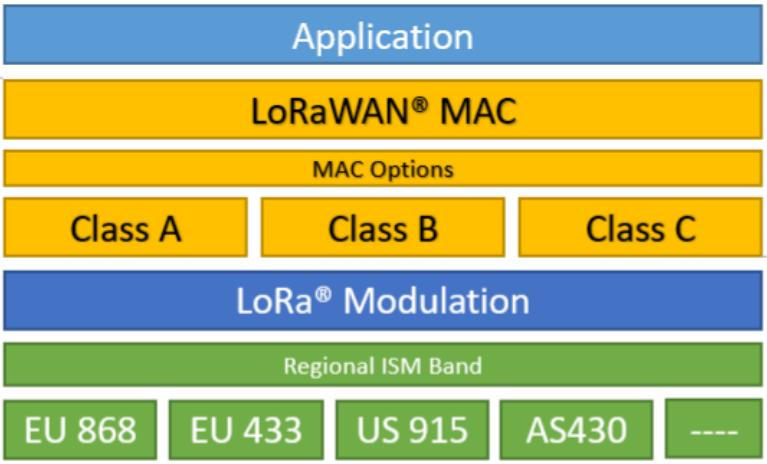

Figure 1.3: LoRaWAN Protocol Stack

Our focus is on the Class A LoRaWAN nodes which transmit based on ALOHA. Nodes wake up and transmit whenever they have packets in their queues.

Chapter 2

State-of-the-art: PC Solutions in LPWANs

Power control in Low-Power Wide-Area Networks (LPWANs) involves various approaches and strategies aimed at managing and optimizing the energy consumption of devices while maintaining reliable communication. There are several algorithms that are proposed to mitigate the interference and minimize the power in the transmitters. We specifically focus on the state-of-the-art PC in LPWANs and a related research publication that propose the minorities and develop the current solution.

As it mentioned, PC algorithms can either centralized or distributed. Sicne the EDs in LPWANs are Microcontroller Units (MCU), their power supply is single-use basttery. SInce EDs are light weight devices, NS perform PC algorithms that has more computational and power resources and havea global view of the network. LoRaWAN also exploits a centralized approach that is implemented at the network server and is activated optionally by the EDs. It collects the information from the history of the transmissions and suggest either to maintain or change the SF and transmit power of the ED. This algorithm is called Adaptive Data Rate(ADR) and it is capable of controlling the ED resources. ADR is beneficial to create a balance between the ED's DR when the channel quality allows higher rates. In this chapter We look in detail the algorithm and the other proposals to develop ADR.

$$p_{rec,i}[dB] = p_i - PL(d_0) - 10n \log_{10}(\frac{d_i}{d_0}) + X_\sigma \tag{2.1}$$

2.1 Adaptive Data Rate (ADR)

ADR works based on the log-distance path loss model in 2.1 and by averaging the received Signal-to-Noise-Ratio (SNR) over a history of EDs transmissions. If the margin between the required SNR and the received SNR is large, it dynamically increase or decrease the DR of the ED. For instance, the gateway receiver received a signal with BW=125KHz and SNR larger than -126dBm. Based on 2.1, sensivity of the receiver in 125KHz is -132dBm for an uplink signal with SF=10 ADR can increase the SF to SF=8 to increase the data rate, or alternatively, it can reduce the transmit power if the SF should be fixed at SF=10.

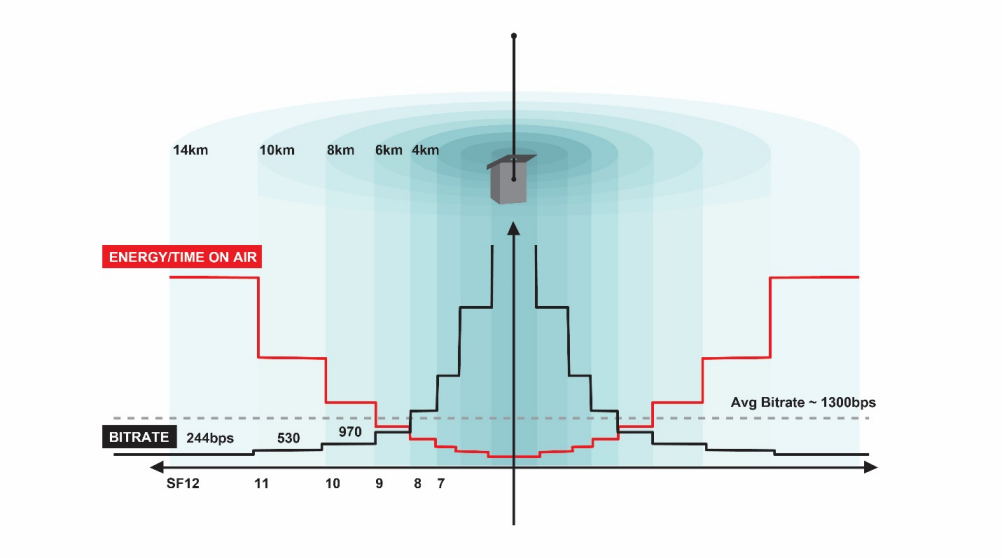

Figure 2.1: Data Rate vs Energy in LoRa [4]

ADR is a centralized algorithm and runs in NS for the static nodes [4]. The algotithm works only if the nodes are static and if they move, the ADR doesn't control the nodes DR. Moreover, it doesn't consider the effect of interference of on the transmissions. CSS is a spread spectrum technique and the codes are not perfectly orthogonal to each other. Since class A devices transmit based on ALOHA, they might interfer with each other. In some scenarios, there is a strong interfere transmitter is close to the gateway. When other nodes transmissions interferes with the interfering node, the distant node may not be heard by the gateway and Capture Effect [2] happens. A research by [5, Yousef A. Al-Gumaei et al] suggest a model to balance the SINR of all the nodes that transmit to a single gateway. It mathematically models the power allocation with a game theory approach were all the nodes try to increase their utility function which is a function of Packet

Success Rate and transmit power. In the next section, we discuss the game theory based approach in more detail and later discuss how it is related to our proposal of power control.

DR/SF	Sensitivity
SF7	-123dBm
SF8	-126dBm
SF9	-129dBm
SF10	-132dBm
SF11	-134.5dBm
SF12	-137dBm

Table 2.1: Receiver Sensitivity at 125KHz [4]

Spreading factor	7	8	9	10	11	12
No of Nodes at Γ	4	7	12	22	39	72
Percentage of Nodes %	2.56	4.49	7.69	14.1	25	46.15

Table 2.2: Percentage of Nodes Assign to Each SF in BE-LoRa

2.2 Game-Theory-Based ADR

Increasing the transmit power increases the PSR and the chance of a successful transmission in the gateway but it comes at the expense of interference for another transmission. Therefore, the objective in this scenario is to maintain an SINR balance for all the nodes communicating with a single gateway. In this kind of power control, the action of nodes, that is, choosing the optimum SF and transmit power affect other nodes decisions and can alter other transmission's qualities. This behavior is closely related to a certain type of non-cooperative game theory that agents play with each other to increase their benefit.

$$f_i(\gamma_i^{SF_k}) = 1 - 0.5e^{-\alpha\gamma_i^{SF_k}} \tag{2.2}$$

$$u_i^{SF_k}(p_i, \mathbf{p}_{-i}^{SF_k}) = \frac{R^{SF_k}(1 - 0.5e^{-\alpha\gamma_i^{SF_k}})}{p_i} \tag{2.3}$$

$$(1 - \frac{(\gamma^{SF_k})^T(M_k - 1)}{G_p^{SF_k}})f'((\gamma^{SF_k})^T)(\gamma^{SF_k})^T = f((\gamma^{SF_k})^T) \tag{2.4}$$

Algorithm 1 Network Server BE-LoRa algorithm

Input: List of LoRa Nodes M, Processing Gain for each SF, $G_p^{SF_k}$, initial Transmission power=14dBm, initial spreading factors=12
Output: The Optimal target SINR for each SF, updated transmit power, updated SF assignment for all nodes

Sort the LoRa Nodes M in descending order of RSSI
for $i = 1$ to M **do**
    Assign node i with an SF based on Table III
end for
Find the Optimal Target SINR for each SF by solving Equation
if $(\gamma^{SF_k})^{opt} < \Gamma_i$ **then**
    $(\gamma^{SF_k})^{opt} = \Gamma_i$
end if
for $i = 1$ to M **do**
    $\gamma_i^{SF_k} = \max(\text{SNR of last 20 frames})$
    if $\gamma_i^{SF_k} > ((\gamma^{SF_k})^{opt} + 1dB)$ **then**
        $p_i = p_i + 1dBm$
    else if $\gamma_i^{SF_k} < ((\gamma^{SF_k})^{opt} + 1dB)$ **then**
        $p_i = p_i - 1dBm$
    else
        Break;
    end if
end for

Power Allocation in LoRaWAN has similar characteristics as the Non-cooperative Game. Nodes communicate based on ALOHA and therefore their transmissions may occur at the same time. Due to the nature of the CSS modulation, nodes which use the different SF have a strong isolation with respect to each other and a large number of nodes choosing different SFs can safely transmit simultaneously. However, if a node transmits while a co-SF node is communicating with the gateway, they interfere with each other and may collide if they have the SIR less than safety margin.[6]

SIR [dB]	7	8	9	10	11	12
7	6	-16	-18	-19	-19	-20
8	-24	6	-20	-22	-22	-22
9	-27	-27	6	-23	-25	-25
10	-30	-30	-30	6	-26	-28
11	-33	-33	-33	-33	6	-29
12	-36	-36	-36	-36	-36	6

Table 2.3: SIR Margin Between The Desired Signal and the Interfering Signal

Configuration	**Bitrate [b/s]**	**Required SNR [dB]**
SF12 / 125 KHz	293	-20.0
SF11 / 125 KHz	537	-17.5
SF10 / 125 KHz	976	-15.0
SF9 / 125 KHz	1757	-12.5
SF8 / 125 KHz	3125	-10.0
SF7 / 125 KHz	5469	-7.5

Table 2.4: LoRaWAN Configuration Table

2.3 Deep Reinforcement Learning-based ADR

Game theory approach is acceptable and it gives an effective solution for the problem. However, it has a few constraints in the long term. In ALOHA, when the number of nodes near a single gateway increases, the probability of interference will potentially increase too. A useful approach for the nodes would be to repeat their transmissions [7]. Repeating the transmission will increase the chance of a successful communication but it again, wastes power on the communicating nodes. We would probably like to train the nodes to learn what is the optimum SF and power level to choose by the Network controller feedback. Moreover, the distributed

approach is useful when the environment changes rapidly e.g the nodes are not fixed for a long time and they will move often due to their nature.

Chapter 3

Reinforcement Learning

Reinforcement learning (RL) is a branch of machine learning that focuses on how agents can make sequential decisions in an environment to maximize a cumulative reward. In RL, an agent learns through trial and error by interacting with its environment, receiving feedback in the form of rewards or penalties for its actions. The goal is to discover a policy that allows the agent to take actions that lead to the most significant cumulative reward R^t over time. RL employs the notion of an exploration-exploitation trade-off, where the agent must balance between exploiting its current knowledge to maximize immediate rewards and exploring new actions that might yield higher long-term rewards. The RL agent starts by making decisions and evaluates its decisions. After each step, it learns probability of moving from state s^t to s^{t+1} by action a^t. As a result of doing the experiment, the agent is given a reward. The transition information is stored in a memory buffer. The key components of RL include:

- **State** $s \in S$ which is a tuple of the environment features that are relevant to the problem
- **Policy** $\pi(s, a)$ that is the probability of taking action a if the current state is s

  $\sum_{a \in A} \pi(s, a) = 1$
- **Action** a^t , the agent take action a and the environment moves from the current state s^t to the s^{t+1}.
- **Transition Probability** The probability that the environment moves from the current state s^t to the s^{t+1} with taking action a

  $P^a_{ss'} = Pr[s^{t+1} = s' \mid S^{(t)} = s, a^{(t)} = a]$
- **Reward** As a result of a transition, the agent receive a reward r^{t+1}

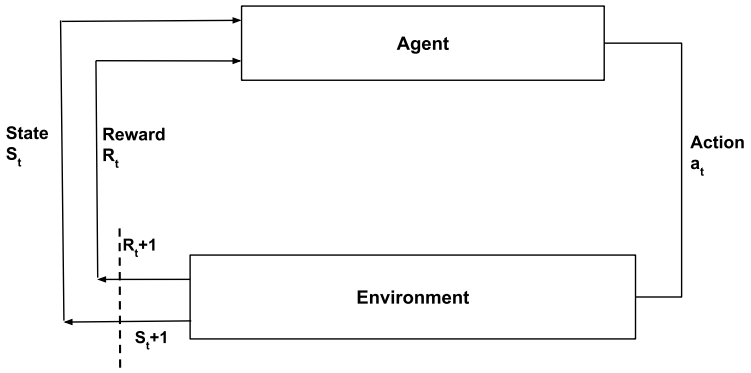

Figure 3.1: Reinforcement Learing Scheme

Reinforcement Learning (RL) encompasses various algorithms designed to enable agents to learn optimal behavior by interacting with an environment. RL algorithms can be

- **Model-based** model-based in which a greedy agent tries to maximize its reward by neglecting the consequence of its actions. In these algorithms, the agent takes its actions by giving a priority to the preferences over the consequences.

  Or

- **Model-free** such that the agent carry out an action multiple times to learn a policy for the optimum rewards.

Some examples of these algorithms are:

- **Policy Gradient Methods** These methods aim to directly learn the policy function, which maps states to actions. Algorithms like REINFORCE (Monte Carlo Policy Gradient) and Proximal Policy Optimization (PPO) optimize policy parameters by adjusting them in the direction that maximizes expected cumulative rewards.
- **Q-Learning** is a model-free RL algorithm used for learning the value of actions in a given state. It aims to find an optimal action-value function, $Q^{\pi}(s, a)$, known as the Q-function, which determines the expected cumulative reward for taking a specific action in a particular state. Through exploration

and exploitation, Q-learning updates Q-values based on observed rewards and transitions.

- **Deep Q-Networks (DQN)** is an extension of Q-learning that utilizes deep neural networks to approximate the Q-function. It employs experience replay and target networks to stabilize learning, allowing for more efficient and stable training on complex environments.

- **SARSA (State-Action-Reward-State-Action)** is another model-free RL algorithm that learns the Q-function by updating action-values based on the current state, action, reward, and the next action chosen following the current policy. It is an on-policy algorithm that updates Q-values during the agent's interactions with the environment.

- **Actor-Critic Methods** Actor-Critic algorithms combine policy-based (actor) and value-based (critic) approaches to learn both the policy and value function simultaneously. Actor-critic methods leverage the advantages of policy gradients and value iteration, offering stable learning and improved sample efficiency. Examples include Advantage Actor-Critic (A2C) and Trust Region Policy Optimization (TRPO).

- **Deep Deterministic Policy Gradient (DDPG)** is an algorithm suitable for continuous action spaces. It combines DQN's experience replay and actor-critic methods to learn a deterministic policy, which can handle continuous action spaces effectively.

- **Multi-Agent Reinforcement Learning (MARL)** deals with multiple agents interacting in an environment. Examples include independent Q-learning for decentralized learning and algorithms like MADDPG (Multi-Agent Deep Deterministic Policy Gradient) for cooperative or competitive multi-agent scenarios.

Most of the RL algorithms are modeled as Markov Decision Processes (MDP). An MDP is a process with fixed number of states. It can evolve from state s to s' in at each step and the probability of transition is fixed. In many cases the agent doesn't have a knowledge of the next transition state or the reward that will receive as a result of that transition. It neither have enough information about the reward of its actions. It explore all the possible states and learns how to maximize its reward by taking actions.

3.1 Q-Learning

Q-learning is a fundamental RL algorithm that finds an optimal action-selection strategy in an MDP [8]. On a model-free setting, the Q-learning agent tries to compute an optimal policy π that maximizes its expected cumulative reward $R^{(t)}$ without knowing the function of the reward or the transition probabilities. Basically, it learns the quality of actions in a given state through exploration and exploitation. The algorithm maintains the tuple (s, a, s', r') its experiments in a table which is called a Q-table.

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[R^{(t)} \mid S^{(t)} = s, a^{(t)} = a] \tag{3.1}$$

Each entry of the table represents the expected cumulative reward 3.2 of taking a particular action in a specific state. By iteratively updating these Q-values based on observed rewards and transitions, the algorithm gradually converges towards an optimal policy that maximizes cumulative rewards over time. This exploration-exploitation trade-off enables the agent to learn which actions are most rewarding in various states, allowing it to make informed decisions while interacting with the environment. Q-learning's simplicity, along with its ability to handle discrete action spaces, has made it a foundational algorithm in reinforcement learning, serving as a basis for more complex and sophisticated methods in the field.

$$R^{(t)} = \sum_{\tau=0}^{\infty} \gamma r^{(t+\tau+1)} \tag{3.2}$$

The reward can be obtained by taking a couple of experiments. Considering the case that moving from the action state (s,a) to s^{t+1} archives a reward R through n experiments. Similarly, the same reward from at (s,a) result can be obtained through less than n experiments. In the second case, the reward can be achieved with a lower number of time steps and have a higher discount factor γ. For example, the anticipated income in one year from now is relative to the income today.

3.2 Deep Q-Learning

Deep Q-Learning (DQL) is an extension of Q-learning that integrates deep neural networks (DNN) to handle high-dimensional state spaces in RL. By leveraging DNNs, it overcomes the limitations of traditional tabular Q-learning, enabling the handling of complex environments. It combines the robustness of Q-learning with the representational power of DNNs to estimate the Q-function in the environments that the state-action space is large and having the look-up table storage is impractical. In our optimization problem which is a wireless environment, the observations

are large and it is not efficient to use the Classic Q-Learning. Therefore, we use another algorithm which estimate the Q-values from DNN which is much more efficient in terms of the storage and convergence.

Deep Q-learning implements two neural networks: The Q-Network which is also known as t train network with parameter $\theta_{train}^{(t)}$ and a target network with parameter $\theta_{target}^{(t)}$. The target network is a copy of the train network and is updated periodically to estimate the Q-values. While the algorithm learns from the environment, the target network converges to the Q-Network by the help of an optimizer that minimizes a loss function 3.3. In the single DQN, both networks co-exist in the same agent and they cooperate with each other, while in the Multi-Agent case, each agent has a copy of the train network $q(s, a, \theta)$ and estimate its own Q-values. The target network gather local information from its environment and estimate the Q-value that gives the maximum reward3.1. The train network $q(s, a, \theta)$ updates the Target Network parameter $\theta_{target}^{(t)}$ every T_u steps. This ensures that the target Q-values used for calculating the loss during training are not as susceptible to rapid changes and fluctuations, resulting in a more stable and effective learning process.

$$L(\theta_{train}^{(t)}) = \sum_{(s,a,r',s') \in D^{(t)}} ({y_{DQN}}^{(t)}(r', s') - q(s, a; \theta_{train}^{(t)}))^2 \tag{3.3}$$

By decoupling the target network's parameters from the Q-network's parameters and updating the target network less frequently, the DQN algorithm becomes more stable and is better able to learn optimal policies in complex environments. The use of a target network is one of the key components that contribute to the success and stability of the DQN algorithm in training deep reinforcement learning models.

Expience Replay

Experience replay is a critical component of the Deep Q-Network (DQN) algorithm that is designed to enhance learning stability and improve sample efficiency by storing and reusing experiences during training. In experience replay, the agent's experiences (s, a, r', s') during its interactions with the environment. Each transition tuple is stored in a replay buffer. In the classical Q-Learning, the agent use each experience immediately after it occurs while DQN saves the transitions into a dataset D, and creates a pool of past experience. While training the Q-network, the algorithm randomly samples mini-batches $D^{(t)}$ of experiences from this replay buffer. This random sampling breaks the temporal correlation between consecutive experiences, which helps to stabilize learning and prevents the model from overfitting to recent experiences.

Optimizer

Optimizer is a specific algorithm to update the parameters of the (Q-network) during training. The optimizer update the weights of the neural network by minimizing a loss derived from the temporal difference error between predicted Q-values and target Q-values.

The choice of optimizer plays a crucial role in efficiently updating the network's weights to minimize the discrepancy between predicted Q-values and target Q-values. Commonly used optimizers in DQN include:

- **Stochastic Gradient Descent (SGD)**used in many neural network applications. SGD updates the network's weights in the direction that reduces the loss, computed as the difference between predicted Q-values and target Q-values, multiplied by the learning rate.

- **Adam** an adaptive learning rate optimization algorithm that combines the benefits of AdaGrad and RMSProp. It dynamically adjusts learning rates for different parameters, making it suitable for non-stationary objectives, which can be the case in reinforcement learning.

- **RMSProp** adjusts the learning rates of each parameter based on the magnitudes of recent gradients. It scales the learning rates by dividing the gradient by the root mean square of past gradients.

- **Adamax** a variant of Adam, Adamax is computationally efficient and has shown good performance in various neural network training tasks. It uses the infinity norm (maximum absolute value) of the gradients in place of the second moment in Adam.

Figure 3.2 descibes architecture of the DQN algorithm and the interaction between the Q-network (train) and the target

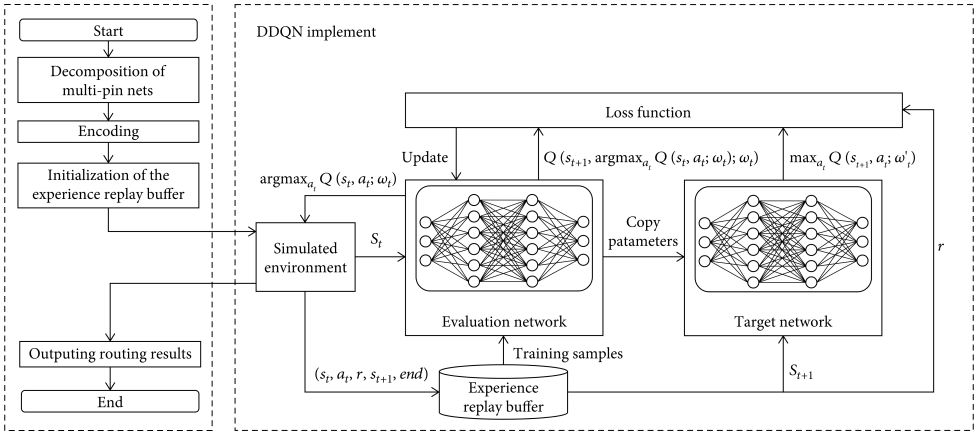

Figure 3.2: Deep Q-network Architecture [9]

$$y_{DQN}{}^{(t)}(r', s') = r' + \lambda \max(q(s', a'; \theta^{(t)}_{train}) \tag{3.4}$$

3.2.1 Multi Agent Q-Learning

Multi-agent deep reinforcement learning involves applying deep reinforcement learning techniques to scenarios where multiple agents interact within a shared environment. Unlike single-agent environments, multi-agent settings introduce complexity due to agents' interactions, resulting in non-stationarity and the emergence of strategic behaviors. These environments encompass diverse applications like multi-agent games, traffic management, collaborative robotics, and resource allocation. Algorithms in multi-agent deep reinforcement learning aim to enable agents to learn decentralized policies that optimize their individual objectives while considering the impact of their actions on the environment and other agents. Techniques such as independent learning, centralized training with decentralized execution, or communication between agents through neural networks facilitate learning in these environments. Handling complex interactions, coordinating actions, and discovering emergent strategies are key challenges addressed by multi-agent deep reinforcement learning, offering promising avenues for solving real-world problems that involve multiple autonomous decision-making entities.

Chapter 4

Conclusion

In this thesis, we propose Deep Reinforcement Learning (DRL) based power allocation for LPWA networks. We studided the state-of-the-art algorithms for power allocation and conclueded that distributed algorithms are promising for IoT standards such as LoRaWAN. We discussed that power control is an optimization problem and can be modelled by mathematical models such as Game Theory and other similar algorithms such as RL has shown to be effective in power allocation in wireless networks.

In the future, we are willing to test the implementation of a Multi-agent Deep Q-Learning approach in real scenarios for LPWANs. Designing a distributed PC approach based on the RL need state-of-the-art technologies and platforms to perform Machine Learning algorithm on the MCU devices.

Bibliography

[1] David Goodman and Narayan Mandayam. «Network Assisted Power Control for Wireless Data». In: 6 (2001), pp. 409–415. DOI: 10.1023/A:1011470315099 (cit. on pp. 1, 2).

[2] Duc-Tuyen Ta, Kinda Khawam, Samer Lahoud, Cédric Adjih, and Steven Martin. «LoRa-MAB: A Flexible Simulator for Decentralized Learning Resource Allocation in IoT Networks». In: *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*. 2019, pp. 55–62. DOI: 10.23919/WMNC.2019.8881393 (cit. on pp. 2, 9).

[3] Bharat S. Chaudhari and Marco Zennaro. «1 - Introduction to low-power wide-area networks». In: *LPWAN Technologies for IoT and M2M Applications*. Ed. by Bharat S. Chaudhari and Marco Zennaro. Academic Press, 2020, pp. 1–13. ISBN: 978-0-12-818880-4. DOI: https://doi.org/10.1016/B978-0-12-818880-4.00001-6. URL: https://www.sciencedirect.com/science/article/pii/B9780128188804000016 (cit. on p. 3).

[4] *What is an Adaptive Data Rate?* https://lora-developers.semtech.com/documentation/tech-papers-and-guides/understanding-adr/ (cit. on pp. 9, 10).

[5] Yousef A. Al-Gumaei, Nauman Aslam, Xiaomin Chen, Mohsin Raza, and Rafay Iqbal Ansari. «Optimizing Power Allocation in LoRaWAN IoT Applications». In: *IEEE Internet of Things Journal* 9.5 (2022), pp. 3429–3442. DOI: 10.1109/JIOT.2021.3098477 (cit. on p. 9).

[6] Claire Goursaud and Jean-Marie Gorce. «Dedicated networks for IoT : PHY / MAC state of the art and challenges». In: *EAI endorsed transactions on Internet of Things* (Oct. 2015). DOI: 10.4108/eai.26-10-2015.150597. URL: https://hal.science/hal-01231221 (cit. on p. 12).

[7] Martin Heusse, Christelle Caillouet, and Andrzej Duda. «Frame Arrival Timing in LoRaWAN: Capacity Increase With Repeated Transmissions and More Channel Attenuation». In: *2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2022, pp. 1048–1054. DOI: 10.1109/PIMRC54779.2022.9977660 (cit. on p. 12).

[8] *Model-Based and Model-Free Reinforcement Learning: Pytennis Case Study.* `https://neptune.ai/blog/model-based-and-model-free-reinforcement-learning-pytennis-case-study` (cit. on p. 17).

[9] Saijuan Xu, Saijuan Xu, and Genggeng Liu. «An Enhanced Deep Reinforcement Learning-Based Global Router for VLSI Design». In: 6.530-8669 (2023), pp. 409–415. DOI: `10.1155/2023/6593938` (cit. on p. 20).