CONCEPTUAL DESIGN OF A TUNNELING RELAY FOR LOW POWER WIDE AREA NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖMER GENÇAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2024

Approval of the thesis:

CONCEPTUAL DESIGN OF A TUNNELING RELAY FOR LOW POWER WIDE AREA NETWORKS

submitted by **ÖMER GENÇAY** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Elif Uysal
Supervisor, **Electrical-Electronics Engineering, METU** _____

Examining Committee Members:

Dr. Elif Tuğçe Ceran Arslan
Electrical-Electronics Engineering, METU _____

Prof. Dr. Elif Uysal
Electrical-Electronics Engineering, METU _____

Dr. Raşit Tutgun
Electrical-Electronics Engineering, Hacettepe University _____

Date:16.08.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Ömer Gençay

Signature :

ABSTRACT

CONCEPTUAL DESIGN OF A TUNNELING RELAY FOR LOW POWER WIDE AREA NETWORKS

Gençay, Ömer
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Elif Uysal

AUGUST 2024, 88 pages

This study addresses the conceptual design of a tunneling relay device intended to overcome the range limitations of traditional Long Range Wide Area Network (LP-WAN) end devices and gateways. The primary objective of the relay is to provide LPWAN connectivity to end devices that suffer from poor channel conditions when attempting to reach the gateway. The relay device operates with a tunneling mechanism, where the uplink packets from end devices are encapsulated and forwarded to the corresponding gateway.

The purpose of this thesis is to conceptually design the tunneling relay device to demonstrate that it enhances network capacity and reduces the network budget by decreasing the number of gateway devices needed.

Keywords: Internet of Things, LoRa, Tunneling Relay, Channel Activity Detection

ÖZ

DÜŞÜK GÜÇLÜ GENİŞ ALAN AĞLARI İÇİN BİR TÜNEL RÖLE KAVRAMSAL TASARIMI

Gençay, Ömer
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Elif Uysal

Ağustos 2024 , 88 sayfa

Bu çalışma, geleneksel Uzun Menzilli Geniş Alan Ağı (LPWAN) uç cihazlarının ve ağ geçitlerinin menzil sınırlamalarını aşmak için tasarlanmış bir tünelleme röle cihazının kavramsal tasarımını ele almaktadır. Rölenin temel amacı, ağ geçidine ulaşmak için zayıf kanal koşullarından muzdarip olan uç cihazlarına LPWAN bağlantısı sağlamaktır.

Röle cihazı, uç cihazlarından gelen yukarı bağlantı paketlerinin kapsüllenip ilgili ağ geçidine iletildiği bir tünelleme mekanizması ile çalışır. Bu tez çalışmasının amacı, röle cihazının ağın kapasitesini artırdığını ve ağda kullanılan ağ geçidi cihazı sayısını azaltarak sistem bütçesini düşürdüğünü kanıtlamak için tünelleme röle cihazını kavramsal olarak tasarlamaktır.

Anahtar Kelimeler: Nesnelerin İnterneti, LoRa, Tünelleme Rölesi, Kanal Aktivite Tespiti

To my family.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Prof. Dr. Elif Uysal, for her invaluable guidance, support, and encouragement throughout this research. Her expertise, patience, and unwavering commitment have been instrumental in shaping this work.

I would like to express my gratitude to my friends and colleagues,from Plan-S, for their support and encouragement. Their friendship and camaraderie have made this journey all the more rewarding.

I am thankful to BTK, the Information and Communication Technologies Authority, and Turkcell for granting me 5G and Beyond Fellowship program.

I would like to thank my little pets Semo, Sengul, Gurbuz, and Feride for their companionship and joy.

Finally, I would like to thank my family for their unwavering support and encouragement, my beloved father Nevzat, my lovely mother Hatice, and my brother Emre for their support and encouragement. My aunt, uncle, and cousins have been a constant source of strength and inspiration. Their love and encouragement have been a guiding light throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
LPWAN	Low Power Wide Area Network
NTN	Non-Terrestrial Networks
IoT	Internet of Things
6G	Sixth Generation
3GPP	3rd Generation Partnership Project
M2M	Machine to Machine
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
ISM	Industrial, Scientific, and Medical
LOS	Line of Sight
MAC	Medium Access Control
ECAD	Enhanced Channel Activity Detection
PCT	The Patent Corporation Treaty
LPWA	Low Power Wide Area
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
Wi-Fi	Wireless Fidelity
MAB	Multi Armed Bandit
UCB	Upper Confidence Bound
TS	Thompson Sampling
CSS	Chirp Spread Spectrum
CPC	Continuous Phase Chirp
PSK	Phase Shift Keying

AWGN	Additive White Gaussian Noise
PN	Pseudo Noise
SAW	Surface Acoustic Wave
SNR	Signal-to-Noise Ratio
SF	Spreading Factor
PHDR	Physical Header
PHYPayload	Physical Payload
CRC	Cyclic Redundancy Check
MIC	Message Integrity Code
FHDR	Frame Header
FPort	Frame Port
DevAddr	Device Address
FCtrl	Frame Control
FCnt	Frame Counter
FOpts	Frame Options
APPSKEY	Application Session Key
MAC	Message authentication Code
NWKSKEY	Network Session Key
WOR	Wake-on-Radio
CAD	Channel Activity Detection
ACK	Acknowledgement
OTAA	Over The Air Activation
ABP	Activation By Personalization
PMU	Power Management Unit
MCU	Microcontroller Unit
RTC	Real Time Clock
MQTT	Message Queuing Telemetry Transport

CID	Command Identifier
DevEUI	Device Extended Unique Identifier
FSPL	Free Space Path Loss
OSI	Open Systems Interconnection
RKHS	Reproducing Kernel Hilbert Space
UART	Universal Asynchronous Receiver-Transmitter
SMD	Surface Mounted Device
SPI	Serial Peripheral Interface
TPTO	Turkish Patent and Trademark Office
WIPO	World Intellectual Property Organization
TUBITAK	Scientific and Technological Research Council of Türkiye

CHAPTER 1

INTRODUCTION

Communication is a fundamental aspect of human existence, encompassing the process of exchanging information between two or more entities. It is a complex phenomenon that can be verbal or non-verbal, encompassing different mediums such as speech, writing, gestures, and signals. The rapid evolution of telecommunications technology over recent decades has ushered in a transformative era characterized by unprecedented connectivity and communication capabilities. In the contemporary digital age, the significance of communication has escalated to unprecedented levels. With the rise of the internet and social media has revolutionized the way people communicate, enabling live interactions regardless of geographical location. Central to this transformation is the advent of Non-Terrestrial Networks (NTN), the Internet of Things (IoT), and the progressive developments encapsulated within sixth-generation (6G) and subsequent 3rd Generation Partnership Project (3GPP) releases.

Moreover, the demand for environmental sensing, has facilitated communication with machines and devices, giving rise to machine-to-machine (M2M) communications. Concurrently, IoT has emerged as a cornerstone of the modern digital ecosystem, interconnecting billions of devices and facilitating real-time data exchange across diverse sectors, including healthcare, agriculture, manufacturing, and smart cities. The proliferation of IoT devices necessitates robust and scalable communication frameworks capable of handling the vast amounts of data generated. In this context, 6G technology, with its enhanced bandwidth, speed, inherent support for sensing, and capacity, serves as a critical enabler for the widespread adoption and effectiveness of IoT solutions.

LoRa (Long Range)[9] is a wireless communication technology characterized by

its long-range and low-power attributes, making it well-suited for IoT applications. LoRa's ability to provide low-power, long-range communication makes it ideal for applications where traditional cellular networks may be impractical or cost-prohibitive. By leveraging the unlicensed spectrum, LoRa enables extensive coverage and deep penetration in challenging environments, such as urban canyons and rural areas. Its integration with 6G and NTN technologies amplifies its potential, creating a cohesive and resilient communication infrastructure capable of supporting diverse IoT applications. LoRa is beneficial for transmitting data over extended distances, thus making it ideal for applications demanding low power consumption and extended battery life.

The LoRa Alliance™ [10] is an open, non-profit association comprising over 500 members, from different markets, dedicated to standardizing Long Range Wide Area Networks (LoRaWAN™) and promoting its widespread adoption. LoRaWAN™ [11] is a protocol developed by the LoRa Alliance™, designed to operate on top of LoRa technology, providing a standardized method for IoT devices to communicate with each other and the cloud by using the LoRa modulation on an unlicensed Industrial, Scientific, and Medical (ISM) band. LoRaWAN™ is noted for its scalability, security, and reliability, making it highly suitable for large-scale IoT deployments.

In a conventional LoRaWAN™ deployment, end devices communicate with a gateway, which subsequently forwards the data to the cloud. However, the effectiveness of LoRaWAN™ is constrained by the physical properties of radio waves, which can be impeded by obstacles such as buildings and trees, posing challenges for end devices in reaching the gateway, particularly in the absence of a direct line of sight (LOS).

To address this limitation, the LoRa Alliance™ has published the TS011-1.0.0 Relay [8] standard, which defines the LoRaWAN™ Relay. This relay serves as an intermediary between end devices and the gateway. The relay device receives uplink packets from the end devices, encapsulates them, and forwards them to the gateway. This mechanism enables end devices to maintain communication with the gateway even when they are beyond the direct range.

By integrating these advanced communication technologies, the potential of IoT applications is significantly enhanced, facilitating smarter and more efficient interconnections between everyday objects and the digital realm. The continuous evolution

of these technologies underscores the critical role of communication in advancing modern technological landscapes.

1.1 Motivation

The current LoRaWAN™ Relay specification [8], while providing a robust framework for wide-area network communications, faces significant limitations in its capacity to support low-end devices and efficiently manage power consumption. These challenges are critical as the Internet of Things (IoT) ecosystem expands, requiring more efficient and scalable solutions to handle a growing number of devices with varying capabilities and power constraints.

One major issue is the limited capacity of end devices within the current LoRaWAN Relay framework. These devices often possess constrained processing power and memory, making it difficult for them to handle the protocol's overhead and data processing requirements. As a result, these devices struggle to participate fully in the network, limiting their utility and the overall network's effectiveness.

Additionally, power consumption remains a significant concern. End devices are typically battery-powered and require optimized energy usage to ensure longevity and sustainability. The existing specification does not adequately address these power constraints, leading to higher energy consumption and shorter device lifespans. This inefficiency is particularly problematic in applications where devices are deployed in remote or difficult-to-access locations, where frequent battery replacement is impractical and costly.

Given these limitations, there is a pressing need to develop an enhanced version of the LoRaWAN™ Relay specification. This new version should focus on increasing device capacity, optimizing energy consumption, and ensuring that even end devices can participate effectively in the network. By addressing these challenges, the enhanced specification will better support the growing and diverse landscape of IoT applications, ensuring more efficient and sustainable network operations.

1.2 Contributions

This thesis work aims to improve the LPWAN network mainly on relay perspective and the gateway perspective. The contributions of this work are as follows:

- A new relay device design that supports more than 10 end devices.
- A new relay device design that reduces power consumption.
- New medium access control (MAC) layer enhancements for the relay device.
- Enhanced Channel Activity Detection (ECAD) mechanism that detects uplink frames from end devices with a low probability of false alarm and a high probability of detection.
- A simulation study that evaluates the performance of the proposed relay device.
- A Patent Corporation Treaty (PCT) patent application for the proposed relay device $PCT/TR2023/051024$.

The new relay device should be able to support more than 10 end devices, reduce power consumption, and improve network capacity by using the proposed MAC layer enhancements. The ECAD mechanism should be able to detect uplink frames from end devices with a low probability of false alarm and a high probability of detection. The simulation study evaluates the performance of the proposed relay device in terms of the probability of detection, the probability of false alarm, the average reward, the average regret, and the instantaneous regret. The PCT patent application is filed for the proposed relay device to protect the intellectual property rights of the proposed relay device.

1.3 The Outline of the Thesis

This thesis includes five chapters. The outline of the thesis is as follows: In Chapter 2, a detailed overview of the related work in the literature is provided. Chapter 3 continues with the design of the tunneling relay and a network with a tunneling relay is

investigated. In Chapter 4, the enhancements on the relay and the network are introduced. In Chapter 5, the results of the performance simulation discussed, and a LoRa end device PCB design is presented. Finally, in Chapter 6, the thesis is concluded, and the possible future works are discussed.

CHAPTER 2

BACKGROUND

2.1 Applications of IoT in Wireless networks

The Internet of Things (IoT) is a transformative technology that has revolutionized the way we interact with the world around us. It is a network of interconnected devices that communicate with each other and exchange data over the internet. The IoT has enabled the creation of smart homes, smart cities, and smart industries, transforming the way we live, work, and play.

The IoT is built on the foundation of wireless networks, which provide the connectivity and communication infrastructure that enables devices to exchange data. Wireless networks are essential for IoT applications, as they allow devices to communicate with each other without the need for physical connections. Wireless networks are characterized by their flexibility, scalability, and mobility, making them ideal for IoT applications that require real-time data exchange and remote monitoring. The number of IoT devices are increasing day by day. At 2027, it is expected that the number of connected IoT devices will be more than 29 billion [12]. The term connected means, connected to each other or connected to Internet. This number includes IoT devices from wireless neighborhood are networks, cellular 6G IoT, wired IoT, low-power wide area (LPWA), cellular IoT, wireless local area networks (WLAN), wireless personal area networks (WPAN). As the frequency spectrum is limited, and the number of wireless devices is increasing, for 6G networks, transmitting the data only when needed is important. Another study in [13] investigates the transmission of a data by considering semantic and goal oriented metrics. Goal Oriented Communication is defined as transmitting data that will be most useful to the point of computation by

the time it gets there [14]. This entails making the data generation as a part of the communication problem.

There are several wireless technologies that are commonly used in IoT applications for LPWAN networks, including Wireless Fidelity (Wi-Fi)[15], Bluetooth[16], Zigbee[17], and LoRa. When the data is need to travel over long distance, it is important to consider protocols or modulations that have robustness for long distances. LoRa is quite capable of transmitting data over long distances. However, the bandwidth and the data rate is restricted to ensure that decoding performance at low received SNR over long distances. LoRa is generally used to transmit status updates. The status update is a data that is a measure of a sensor, or an actuator controlled with low data sizes.

Transmitting a status update is not a straightforward process for same applications. Since the power is limited for a sensor, because it is placed an area that has no power line, decision to send the data is important for the sake of power consumption. In that case semantic metrics enables measuring the importance of the data. With semantic measures, a machine would know it is really necessary to transmit the data or not.

Moreover, it is also important to make energy efficient data transfer to make sure the transmitter lives longer. Energy efficiency in data transmission has a big place in the literature. Starting with the formulation in [18], which considers, the energy minimization problem for a node in a wireless data network to transmit packetized information within a given amount of time is considered. In [19], the work in [18] is expanded and minimizing the energy consumption of a node in a point-to-point link is considered. The packet transmission times and power levels are varied to find the optimal schedule for transmitting data packets within the given amount of time.

These fundamental studies layed out the basics of low-energy and energy-harvesting machine type communication. These will be important concepts in 6G, someting denoted as "zero-energy communication"

2.2 Semantics

Semantics usually refers to the meaning that is conveyed by a message. It is the information that is extracted from the message, and it is the information that is relevant to the receiver. However, in the context of communication networks, [14] defined semantics as not the meaning but the purpose of the data as it relates to the goal of the communication. The given end-to-end architecture enables the generation of just the right amount of data and transmission of the right content to the right place at the right time. It is stated that this will be achieved by redesigning the entire process of information, generation, transmission and usage of the data.

Moreover, in [20], the "generate-at-will" model is studied. It is assumed that the source node has access to the channel state whether is busy or idle. Since the status update is waiting in the queue to be transmitted, it became stale as time goes. The study suggests that it is better not to generate updates when the channel is busy, and this eliminates the waiting time in the queue.

For another perspective, the channel state detection can be costly if the source node is designed to be energy efficient. It is also important to adjust the channel scan times of the source node. This brings the multi-armed bandit problem to the scene.

2.3 Multi Armed Bandit Problems

The multi armed bandit (MAB) problem is a classic problem in probability theory, statistics, and machine learning that involves finding the optimal strategy for allocating resources among multiple options. It is based on the concept of a gambler who is faced with a row of slot machines, each with a different payout probability. This is called the explore-exploit dilemma. Where the exploration is to find the best machine to play, and the exploitation is to play the best machine found so far. The gambler must decide which machine to play in order to maximize his winnings over time. Moreover, the gambler must also decide when to change the machine to play.

There are two success measures for a MAB problem, one is the gained reward until a specific time and the other is the payoff until reaching a reward. The reward is the

value that is gained by playing a machine, and the payoff is the value that is gained by playing a machine until a specific time.

The multi armed bandit problem has many applications in real-world scenarios, such as clinical trials, online advertising, and network routing. There is no generalized, unique solution to any multi armed bandit problem, since the problem is highly dependent on the context and the constraints of the problem.

There are several algorithms that have been developed to solve the multi armed bandit problem for different constraints. When defining the context and constraints, the reward function, the exploration-exploitation trade-off, and the regret are the main factors to consider.

In [21], the Upper Confidence Bound (UCB) algorithm is introduced to solve the multi armed bandit problem. The UCB algorithm is based on the principle of optimism in the face of uncertainty, which means that the algorithm selects the arm that has the highest upper confidence bound. The UCB algorithm is known to be one of the most effective algorithms for solving the multi armed bandit problem.

The empirical reward for any source i at time instance t is defined as $\hat{\mu}_{i,t}$, and it is shown in Equation (2.1).

$$\hat{\mu}_{i,t} = \frac{\sum_{s=1}^{t} R_{i,s}}{n_{i,t}} \tag{2.1}$$

Where $R_{i,s}$ is the reward gained by playing the source i at time s, and $n_{i,t}$ is the number of times the source i is played until time t.

The UCB algorithm is based on the principle of optimism in the face of uncertainty, which means that the algorithm selects the arm that has the highest upper confidence bound. The upper confidence bound for any source i at time instance t is defined as $U_{i,t}$, and it is shown in Equation (2.2).

$$U_{i,t} = \hat{\mu}_{i,t} + \sqrt{\frac{2\ln(t)}{n_{i,t}}} \tag{2.2}$$

The UCB algorithm is given in Algorithm 1.

Algorithm 1 Upper Confidence Bound (UCB) Algorithm

1: **Input:** Number of sources N, Time horizon T
2: **Initialization:** Play each source once
3: **for** $t = N + 1, T$ **do**
4: Play source i that maximizes $U_{i,t}$
5: Observe reward $R_{i,t}$
6: Update empirical reward $\hat{\mu}_{i,t}$
7: Update upper confidence bound $U_{i,t}$
8: **end for**

In [22], the Thompson Sampling (TS) algorithm is introduced to solve the multi armed bandit problem. It is a Bayesian approach to the MAB problem. The main difference of the TS from UCB is that it takes a frequentist approach. Moreover, it is the first algorithm proposed for MAB problems.

The TS algorithm is based on the principle of sampling from the posterior distribution of the reward for each arm. Consider that the sources give reward with Bernoulli distribution, and let p_i is the success probability for source i. The distribution of p_i is Beta distribution[23] with prior estimate of p_i is $\hat{\nu}_i$, and Γ is the Gamma function. The Equation (2.3) tries to converge the posterior distribution of the reward for each arm by sampling from the Beta distribution.

$$f\left(\hat{\nu}_i(t); \alpha_i; \beta_i\right) = \frac{\Gamma\alpha_i + \beta_i}{\Gamma\alpha_i \Gamma\beta_i} \hat{\nu}_i(t)^{\alpha_i - 1} (1 - \hat{\nu}_i(t))^{\beta_i - 1} \tag{2.3}$$

For Thompson Sampling, the parameters α_i and β_i represents the success and failure counts of source i, where $\alpha_i > 1$ and $\beta_i > 1$. For each iteration, the posterior Beta distribution is sampled for each source, and the source with the highest sample is played. The exploration and exploitation process is balanced by the Beta distribution. Each success increases the α_i and each failure increases the β_i. The initial assumtion is that $\alpha_i = 1$ and $\beta_i = 1$ for each source, and the algorithm is given in Algorithm 2.

Finding the best arm can be considered as the shortest path problem. There is a Greedy algorithm which can be traced back to the early Dijkstra algorithm [24]. Greedy algorithm is also advantageous for its simplicity and efficiency. The Greedy

Algorithm 2 Thompson Sampling (TS) Algorithm

Input: Number of sources N, Time horizon T
Initialization: $\alpha_i = 1$, $\beta_i = 1$ for each source i
for $t = 1, T$ **do**
  Sample $\hat{\nu}_i(t)$ from Beta distribution $f\left(\hat{\nu}_i(t); \alpha_i; \beta_i\right)$
  Play source i that maximizes $\hat{\nu}_i(t)$
  Observe reward $R_{i,t}$
  if $R_{i,t} = 1$ **then**
    $\alpha_i = \alpha_i + 1$
  else
    $\beta_i = \beta_i + 1$
  end if
end for

algorithm is reviewed in [25] to reveal the theoretical and practical challenges of the Greedy algorithm.

The Greedy algorithm has several extensions. One of the extensions is the ϵ-Greedy algorithm[26]. The ϵ in the name states for a probability value. The algorithm adjust its exploration and exploitation distribution based on the ϵ value. The Greedy algorithm can also be extended by initializing a prior rewards for each arm to control the exploration process at the first place. For ϵ-Greedy algorithm, let Q_i be the prior reward for source i, and N_i be the number of times source i is played.The ϵ-Greedy algorithm is given in Algorithm 3.

2.4 LoRa

LoRa is a cutting-edge wireless communication technology characterized by its long-range capabilities and low-power consumption, making it particularly well-suited for IoT applications. By leveraging unlicensed radio spectrum, LoRa enables the transmission of data over extensive distances, making it an optimal choice for applications that prioritize energy efficiency and prolonged battery life.

LoRa is a proprietary technology developed by Semtech Corporation [9]. The mod-

Algorithm 3 ϵ-Greedy Algorithm

Input: Number of sources N, Time horizon T, ϵ
Initialization: $Q_i = 0$, $N_i = 0$ for each source i
for $t = 1, T$ **do**
    With probability $1 - \epsilon$, play source i that maximizes Q_i
    With probability ϵ, play a random source
    Observe reward $R_{i,t}$
    Update $Q_i = \frac{R_{i,t} + Q_i \cdot N_i}{N_i + 1}$
    Update $N_i = N_i + 1$
end for

ulation is based on Chirp Spread Spectrum (CSS) technology. The main difference of the CSS technology from other spread spectrum technologies is that it does not use pseudo-random codes to spread the signal. Instead, it uses chirp signals to spread the signal in the frequency domain. This allows the signal to be received even if it is below the noise floor.

CSS technology is based on the principle of linear frequency modulation. The signal is modulated by varying the frequency of the carrier signal over time. This results in a signal that is spread over a wide frequency band, making it less susceptible to interference and fading.

Before CSS, using chirp signals for data communications is first introduced in [27]. In [28], the continuous phase concept is applied to chirp signals, theoretical error rate bounds are established for continuous phase chirp (CPC) signalling systems with coherent reception, and it found that CPC performs as well as or better than antipodal phase shift keying (PSK) in an additive white Gaussian Noise (AWGN). The optimum non-coherent receiver is obtained, and error rate is analyzed for CPC in [29]. When time-frequency region is limited, pseudo-noise (PN), and PSK is combined and denoted as PN-chirp in [30]. When spectrum is allocated more than before as the time goes, spread spectrum CDMA is suggested to share the spectrum with existing narrowband users in [31].

Chirp signals for spread-spectrum communication were tested in indoor and industrial environments in [32]. Surface acoustic wave (SAW) chirped delay lines are presented

to achieve high bit rate and $\pi/4 - DQPSK$ modulation were used to eliminate the difficulties of carrier recovery of traditional QPSK.

After the introduction of LoRa to the academic world and wireless market, research on LP-WAN networks emphasized the importance of LoRa.

In [33], LoRa is compared with its competitors in the Low Power Wide Area Networks (LPWAN) market Sigfox[34] and Ingenu[35]. The LoRa coverage in Padova, Italy is investigated as a use case. Moreover, different LP-WAN network technologies namely, SigFox, LoRa, Ingenu, Telensa[36], Qowisio[37] are compared with each other and 3GPP Cellular IoT. Also in [38], the performance of CSS is compared with SigFox in coverage and coexistence with other competing signals in the same band aspects. In [39] LP-WAN technologies are compared in case of capacity and collision. The results show that NB-IoT outperforms the others. However, it is important to consider that an advantage of LoRa is its ability to operate in SRD bands.

The limits of LoRa in indoor scenarios is inspected in [40]. More detailed review on the LoRaWAN™ limits is given in [41]. The limits are considered in network size with limited duty cycle, reliability, and dense network capacity aspects. Also, LoRaWAN™ use cases such as metering, real-time monitoring, smart transportation and logistics, smart city application, and video surveillance are investigated.

The BER performance of the LoRa modulation is studied in [42] by considering the spreading effect, the channel coding, and the signal-to-noise ratio (SNR). A signal attenuation model is introduced in [43] for a real life scenario in Dakar Region, Senegal.

As seen in Figure 2.1, the signal is spread over a wide frequency band, which allows it to be received through a despreading operation even if it is below the noise floor. This makes it ideal for long-range communication, as the signal can be received even in the presence of noise and interference. The chirps are spread all over the determined bandwidth. The spreading factor (SF) determines the duration of the chirp signal. The higher the spreading factor, the longer the duration of the chirp signal. This allows the signal to be received even in the presence of noise and interference.

The advantage of the LoRa CSS is that timing and frequency offsets are equal for the

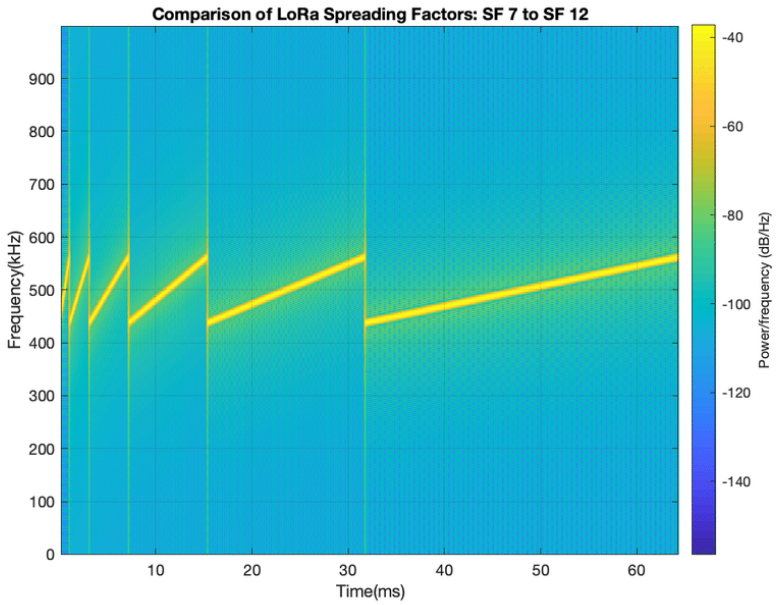

Figure 2.1: Chirp Spread Spectrum Signal for different Spreading Factors [5]

transmitter and the receiver. This greatly reduces the complexity of the receiver.

The message is chipped at high chip rate and modulated onto the chirp signal. The Equation (2.4) shows the relation between the data bit rate, symbol rate and chip rate for LoRa modulation [2].

$$R_b = \frac{SF}{\left[\frac{2^{SF}}{B}\right]} \quad \text{bits/sec} \tag{2.4}$$

Where SF is the spreading factor, and B is the bandwidth of the signal. The spreading factors are usually between 7 and 12, and the bandwidths are between 125 kHz and 500 kHz for EU868 region of LoRaWAN™Regional Parameters[6].

2.5 LoRaWAN

LoRaWAN™ is a protocol developed by the LoRa Alliance™ to run on top of LoRa modulation and provide a standard way for IoT devices to communicate with each other and with the cloud. LoRaWAN™ is scalable, secure, and reliable, making it

ideal for large-scale IoT deployments. It defines the communication protocol and system architecture for a LoRaWAN™ network. There are three classes of LoRaWAN™ end devices, which are Class A, Class B, and Class C. Each class has different characteristics and capabilities.

The components of a LoRaWAN™ network are end devices, relays, gateways, network servers, and application servers. The end devices are the devices that communicate with the network. They can be sensors, actuators, or any other type of device that needs to send data to the cloud. The relay devices are the devices that forward the data from the end devices to the gateways. They act as a bridge between the end devices and the gateways. The gateways are the devices that receive the data from the end devices and forward it to the network server. They act as a bridge between the end devices and the network server. The network server is the device that manages the network and controls the communication between the end devices and the gateways. It is responsible for routing the data to the appropriate application server. The application server is the device that processes the data received from the end devices and performs the necessary actions. It is responsible for storing the data and providing it to the user.

2.5.1 LoRaWAN™ Classes

2.5.1.1 Class A

Class A devices are the most common type of LoRaWAN™ end devices. They have the lowest power consumption and the only longest battery life. They are designed to conserve power by only transmitting data when necessary.

Class A devices have two receiving windows, which are used to receive downlink data from the network server just after the uplink transmission. The receiving windows are opened for a short period of time (from 1 second to 15 seconds [7]) after the uplink transmission, and the device listens for any downlink data from the network server.

If there is no downlink data, the device goes back to sleep mode to conserve power. Class A devices are ideal for applications that require low power consumption and

long battery life.

2.5.1.2 Class B

Devices that use Class B mode implement the Class A functionality and on top of this, can receive downlink data at scheduled times. They have additional receiving windows, which are opened and managed by the network server with a time synchronization mechanism by using the beacon signals.

The network server sends a beacon signal to the Class B devices at regular intervals, which is used to synchronize the receiving windows. The receiving windows are opened at the scheduled times, and the device listens for any downlink data from the network server. Class B devices are ideal for applications that require low power consumption and scheduled downlink data reception at the same time.

2.5.1.3 Class C

Class C devices implement the Class A functionality, and on top of this, they also have continuous receiving windows. The receiving windows that are always open, allow them to receive downlink data at any time. Class C devices are ideal for applications that require continuous downlink data reception, without concern for the power consumption.

2.5.2 LoRaWAN™Frame

LoRaWAN™ frame is the basic unit of data transmission in a LoRaWAN™ network, it is defined in [6]. On air frame format is shown in Figure 2.2.

The LoRaWAN™ frame consists of the following fields: preamble, synchronization word, physical header (PHDR), physical payload (PHYPayload), and cyclic redundancy check (CRC).

The preamble is used to synchronize the receiver with the transmitter. It is a sequence of alternating 0s and 1s that is used to detect the start of the frame. The synchroniza-

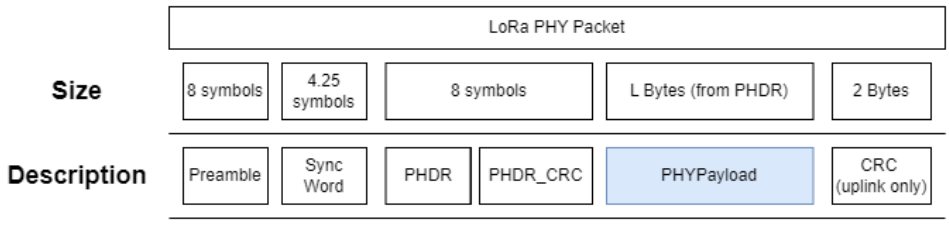

Figure 2.2: LoRaWAN™ Packet Format [6]

tion word is used to synchronize the receiver with the transmitter. It is a fixed pattern that is used to detect the start of the frame. The PHYPayload is the actual data that is transmitted in the frame. It can be up to 255 bytes long and can contain any type of data. The CRC is used to detect errors in the frame. It is a fixed pattern that is used to detect errors in the frame.

In the LoRaWAN™ frame, the PHYPayload field is the most important field, it is defined in LoRaWAN™ Medium Access Control (MAC) Layer Specification [7]. It contains the actual data that is transmitted in the frame, and it is shown in Figure 2.3.

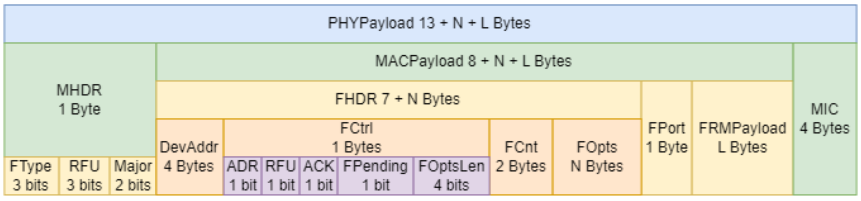

Figure 2.3: PHYPayload Format [7]

The PHYPayload field consists of the following fields: MAC header (MHDR), MACPayload, and message integrity code (MIC). The MHDR field is used to indicate the type of frame and version of the frame format. It is a fixed pattern that is used to detect the start of the frame.

The MACPayload field is the actual data that is transmitted in the frame. MACPayload includes frame header (FHDR), frame port (FPort), and frame payload (FRMPayload). The FHDR field is used to indicate the type of frame and identify the end device. It includes the device address (DevAddr), frame control (FCtrl), frame counter (FCnt), and frame options (FOpts). The FPort is used to differentiate different types of application data.

The FRMPayload is the actual data that is transmitted in the frame. It is encrypted using the application session key (AppSKey).

The MIC is a message authentication code (MAC) that is used to verify the integrity of the frame. It is calculated and verified using the network session key (NwkSKey).

2.5.3 LoRaWAN™ Gateway

LoRaWAN™ gateway is the device that receives the data from the end devices and forwards it to the network server. It acts as a bridge between the end devices and the network server. The gateway is responsible for receiving the data from the end devices, demodulating the data, and forwarding it to the network server. It is also responsible for transmitting the downlink data from the network server to the end devices. The gateway is usually connected to the network server using an IP-based connection, such as Ethernet or Wi-Fi.

The hardware components of a LoRaWAN™ gateway are the radio module, the microcontroller, and the network interface. The radio module usually includes two front end modules, each have 4 channels to receive the data from the end devices. The two front end modules are driven by a baseband processor, which is responsible for demodulating the data. Between the gateway and the network server, there is a packet forwarder that is responsible for forwarding the data to the network server.

2.5.4 LoRaWAN™ Relay

A LoRaWAN™ relay is a device that forwards the data from the end devices to the gateways. There is a specification TS002-1.0.0 [8] that defines the LoRaWAN™ relay device. This specification describes a new protocol defined for the relay devices, which requires updates on both end device side and the network server side. A new frame Wake on Radio (WOR) is defined for the relay devices, which is used for waking up the relay device. The purpose is to keep the power consumption of the relay device low.

The main difference of the WOR frame with the regular frame is that the pream-

ble size, which is longer than the preamble, which is up to 1 second, of a regular LoRaWAN™ frame. The relay device is sleeping most of the time, and it wakes up periodically to listen for the WOR frame by applying channel activity detection (CAD) mechanism. More information will be provided in Chapter 3.

2.5.5 LoRaWAN™ Network Server

The network server, which is defined in LoRaWAN™ Backend Interfaces [44], is a software component that behave as a core network for a LoRaWAN™ network. It is responsible for device activation, frame routing, gateway registration, and frame decryption.

The network server is connected to the gateways by IP based connections. It is also responsible for sending the received sensor data to the application server.

2.5.6 LoRaWAN™ Application Server

The application server, which is defined in LoRaWAN™ Backend Interfaces [44], is a software component that is responsible for processing the data received from the end devices. It is responsible for storing the data, performing the necessary actions, and providing the data to the user. The application server is connected to the network server by IP based connections.

CHAPTER 3

TUNNELING RELAY

3.1 Transferring Needed Data Over Long Distances

The novel work in this thesis started by asking the following question: How to transfer the needed data over long distances by using low-power? To answer this question, there is a need to define the requirements of the network:

- The network should be able to transfer data over long distances (>1km).
- The network should be able to transfer data by using low-power (i.e. an end device can last at least 5 years).
- The network should be scalable.
- The network should satisfy end-to-end security.

In order to satisfy the first requirement, there are several options such as using high power, using modified hardware or antennas, or transferring data over multi hops. Using modified hardware or antennas is not a cost-effective solution.

Therefore, transferring data over multi hops is the best option to satisfy the first two requirements if the definition of long distance cannot be satisfied by a single hop.

For scalability, the network should be able to add new nodes easily and the network should be able to work with numerous nodes.

For the end-to-end security requirement, the network should be able to provide security, integrity, and availability.

When the low end-to-end delay requirement is taken into consideration, the network should be able to transfer data with low latency.

There is one more question to ask: When to send the data, how to be sure that the data should be sent? The answer of when to send data can be answered by defining semantic parameters that define the need of the data.

3.1.1 Multi Armed Bandit Approach to An LPWAN Network[1]

A similar problem is solved in [1] with applying MAB problem on system model and introducing a reward when useful data is transferred.

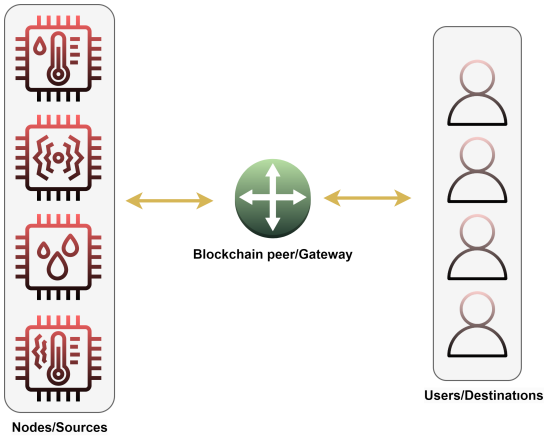

Figure 3.1: System Model of the MAB Problem

The system model for [1] is given in Figure 3.1. There are N sources and M destinations in the system. The sources have data to send to the destinations over a gateway. The gateway schedules the sources to send the data to the destinations.

$$R_i(t) = \begin{cases} 1, & \text{if semantic effectiveness is satisfied} \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

The reward function defined in Equation (3.1) states that if the semantic effectiveness is satisfied, the reward is earned, otherwise there will be no reward.

$$\max_{\pi \in \Pi} \quad \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} \mathbb{E}[R_i(t)] \tag{3.2}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} u_i(t) \leq k, \forall t, \tag{3.3}$$

The optimization problem defined in Equation (3.2) states that the goal of the policy π is to maximize the time-average expected reward $R_i(t)$ for all arms at time instance t by scheduling the transmitters that have data that is semantically effective. The constraint states that the number of transmitters $u_i(t)$ that can be scheduled at time t is limited by k.

Next, we will define the relay idea and the relay standard that is defined in LoRaWAN™, before introducing the tunneling relay design.

3.2 The Relay Idea

In LoRaWAN™, there is a relay standard that satisfies the multi-hop requirement. However, the relay standard has some limitations such as supporting only up to 10 end devices and requiring an additional process to communicate with an end device which brings additional power consumption and software update requirement on the end devices that are already deployed. We overcome these limitations, with a new relay device design.

Before explaining the new relay device design, the already defined relay standard should be explained in detail to make a comparison between the existing relay standard and the new relay device design.

3.3 The Existing Relay Standard

The existing relay standard is defined in [8]. The relay device receives uplink packets from the end devices, encapsulates them, and forwards them to the gateway. The relay device can support up to 10 end devices. It is assumed that the relay device is powered by a battery, therefore, the relay device should be designed by considering low-power consumption.

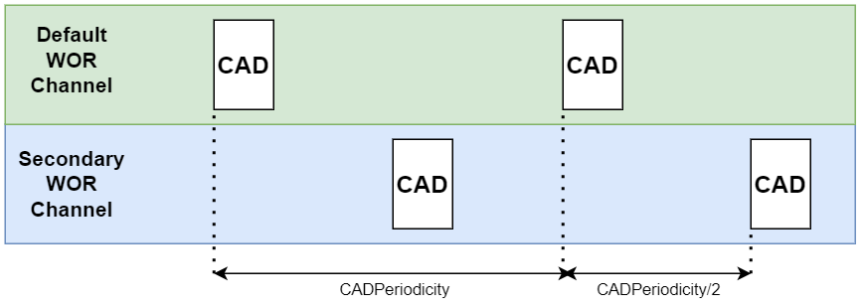

Figure 3.2: Channel Activity Detection On A Channel [8]

Similar to an end device operation, the relay device is required to sleep most of the time. In addition to the end device operation, it wakes up to perform CAD and checks if there is a preamble on the predefined mandatory relay-access channel and optional secondary channel periodically as given in Figure 3.2. If a LoRa preamble is detected, the relay device tries to demodulate the received signal and checks if the received signal is a valid LoRaWAN™WOR frame.

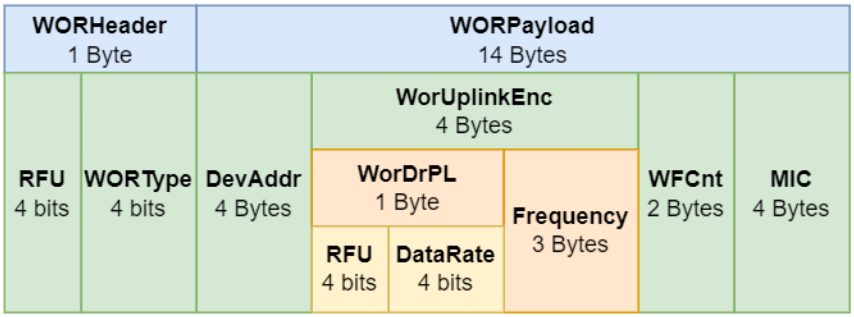

Figure 3.3: Wake-On-Radio Class A Uplink Frame [8]

There are two types of WOR frames coming from the end devices which are the WOR Relay Join-Request and WOR Class A Uplink frames. If the end device is joined the network and activated somehow before, it directly sends the WOR Class A Uplink frame, otherwise, it sends the WOR Relay Join-Request frame to initiate the relay join procedure.

The WOR Class A uplink frame includes 1 Byte $WORHeader$ and variable length $WORPayload$. The $WORHeader$ includes the $WORType$ and reserved RFU fields. The $WORType$ can be a relay join-request, relay Class A uplink, or proprietary. The WOR Class A Uplink format is shown in Figure 3.3.

The $WORPayload$ includes 4 Bytes $DevAddr$, 4 Bytes $WorUplinkEnc$, 2 Bytes $WFCnt$, 4 Bytes MIC. The $DevAddr$ is the device address of the end device. The $WorUplinkEnc$ is the encrypted uplink frame. The $WFCnt$ is the frame counter of the end device. The MIC is the message integrity code of the uplink frame. The $WorUplinkEnc$ is the encrypted form of the $WorUplink$ field with $WorSEncKey$, and it includes data rate and frequency information about the upcoming uplink frame from the end device. The $WorUplink$ field is encrypted by using AES-128 in CTR mod with $WorSEncKey$, which is derived from the $NwkSKey$ given on the activation.

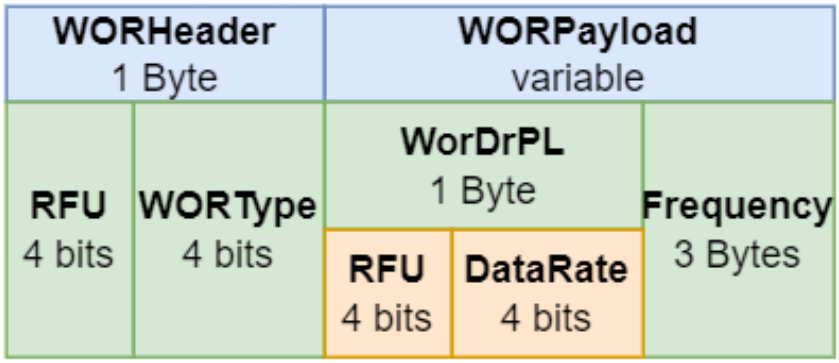

Figure 3.4: Wake-On-Radio Join Request Frame [8]

The WOR Relay Join-Request frame also include $WORHeader$ and $WORPayload$ fields as seen in Figure 3.4. The $WORPayload$ includes data rate and frequency information about the upcoming join-request just after the relay join procedure.

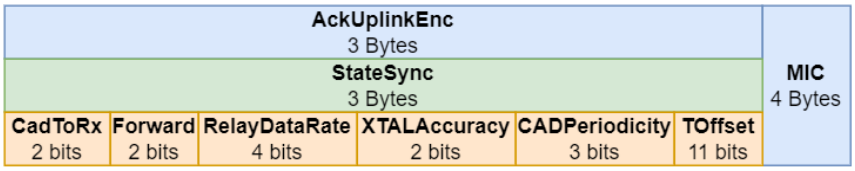

Figure 3.5: Wake-On-Radio Uplink ACK Frame [8]

The relay returns a WOR acknowlegment (ACK) Uplink frame to the WOR Class A Uplink frame. The WOR ACK Uplink frame structure is given in 3.5 The WOR ACK Uplink frame includes 3 Bytes $ACKUplinkEnc$, 4 Bytes MIC.

The $ACKUplinkEnc$ is the encrypted form of the $ACKUplink$ field with $WorSEncKey$,

and it includes the $StateSync$ field.

The $StateSync$ field is the synchronization information about the relay device and the end device. It includes $CadToRx$ field which indicates the time between the CAD and the RX window opening,

$Forward$ field indicates the relay forward limitations, $RelayDataRate$ field is the data rate of the relay device, $XTALAccuracy$ field is the accuracy of the crystal oscillator of the relay device to protect the synchronization between the end device and the relay, $CADPeriodicity$ field is the periodicity of the CAD operation of the relay device, $TOffset$ is the time offset between the start and end of the WOR frame preamble, in milliseconds. MIC is the message integrity code calculated by using $WorSIntKey$ and $ACKUplinkEnc$ using AES-128 CMAC algorithm.

The received uplink packet from the end device is sent to the network server by using a Medium Access Control (MAC) command, which is ***ForwardUplinkReq***. The MAC command consists of 6 Bytes of metadata, and the uplink payload, the packet format is shown in Figure 3.6.

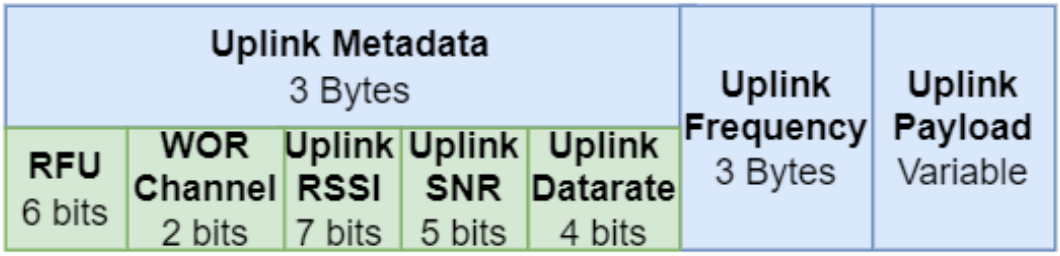

Figure 3.6: ForwardUplinkReq MAC Command format [8]

The Figure 3.6 is the $FRMPayload$ of a LoRa packet. $UplinkMetadata$ includes 5 fields, which are, $UplinkDatarate$, $UplinkSNR$, $UplinkRSSI$, $WORChannel$, and RFU. The $UplinkDatarate$ is the data rate of the uplink frame from the end device. The $UplinkSNR$ is the signal-to-noise ratio of the uplink frame from the end device. The $UplinkRSSI$ is the received signal strength indicator of the uplink frame from the end device. The $WORChannel$ is the channel that the relay device received the uplink frame from the end device. The RFU is reserved for future use.

The activation of the end devices can be done by two different ways which are the

Over-The-Air Activation (OTAA) and the Activation-By-Personalization (ABP). The ABP is a simple activation method and commonly used for private LoRa network. On the other hand, the OTAA is a more secure activation method and commonly used for public LoRa networks. The join procedure for a relay session adds more complexity and power consumption to the end devices.

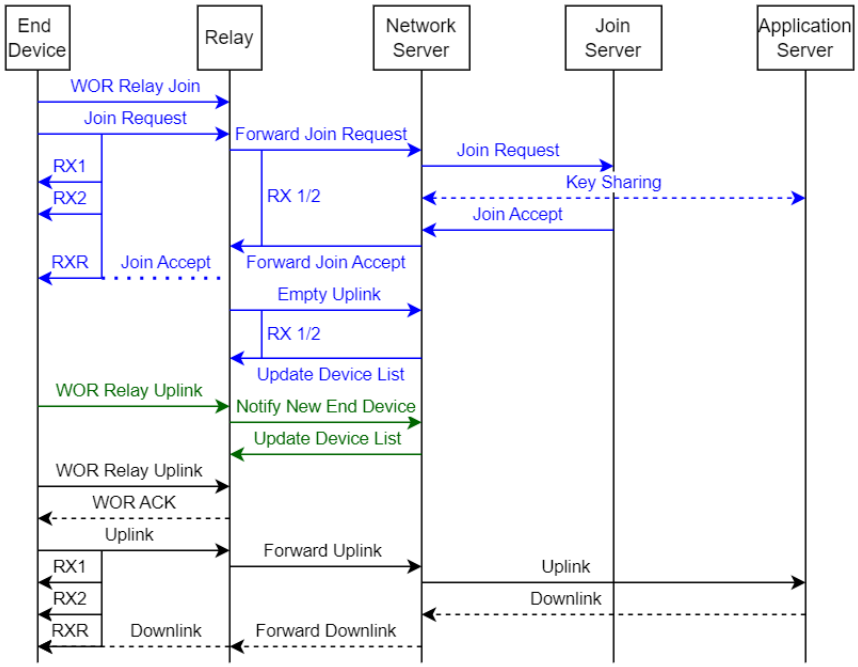

Figure 3.7: Relay Sequence Diagram [8]

As the sequence diagrams given in Figure 3.7, if an end device wants to send data over a relay, it should send three additional uplink frames to the end device and opens four more receiving windows to receive the downlinks from the relay and the network server over the relay for a join request. The join procedure sequence is stated with color blue in the Figure 3.7.

If the end device already joined the network, it should send at least one or two additional uplink frames to the relay. The additional uplink is stated in green in the Figure 3.7. Moreover, the end device opens two more receiving windows to receive the downlink transmissions from the relay and the network server over the relay for a Class A uplink. The default receiving windows are called $RX1$ and $RX2$. With the relay extension, there is a new reception window called RXR which is opened for the network server response to the end device over the relay.

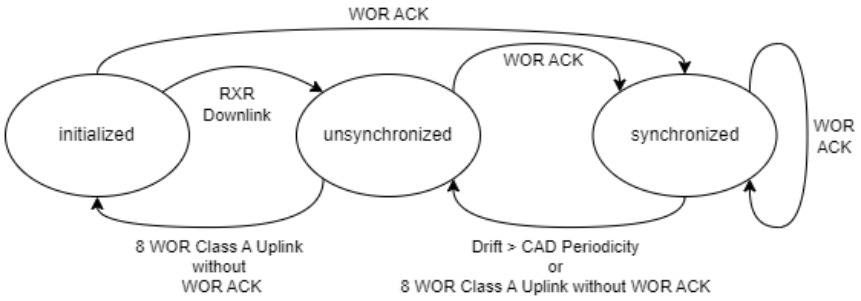

Figure 3.8: LoRaWAN™ Relay State Transitions

There are three synchronization states defined between end device and the relay device. The state transitions are given in Figure 3.8. The first state is the "initialized" state, where the end device is unaware of the relay CAD periodicity, relay crystal oscillator frequency accuracy, active relay channels, default relay channel, etc.

The second state is the "synchronized" state, where the end device is synchronized with the relay device.

The transition from the "initialized" state to the "synchronized" state is done by the relay device sending a WOR ACK frame to the end device.

The third state is the "unsynchronized" state, where the end device is not synchronized with the relay device.

The transition from the "synchronized" state to the "unsynchronized" state is done if the clock drift of the end device is bigger than the CAD periodicity or the end device sent 8 WOR frames without receiving any WOR ACK frame.

The transition from the "unsynchronized" state to the "synchronized" state is done by the relay device sending a WOR ACK frame to the end device.

The transition from the "unsynchronized" state to the "initialized" state is done if the end device sent 8 WOR frames without receiving any WOR ACK frame.

The transition from the "initialized" state to the "unsynchronized" state is done if the end device received downlink in the relay receiving window.

3.4 The Tunneling Relay Design

The tunneling relay is designed similar to the existing relay standard but with some modifications on the software. The reference design is given in Figure 3.9

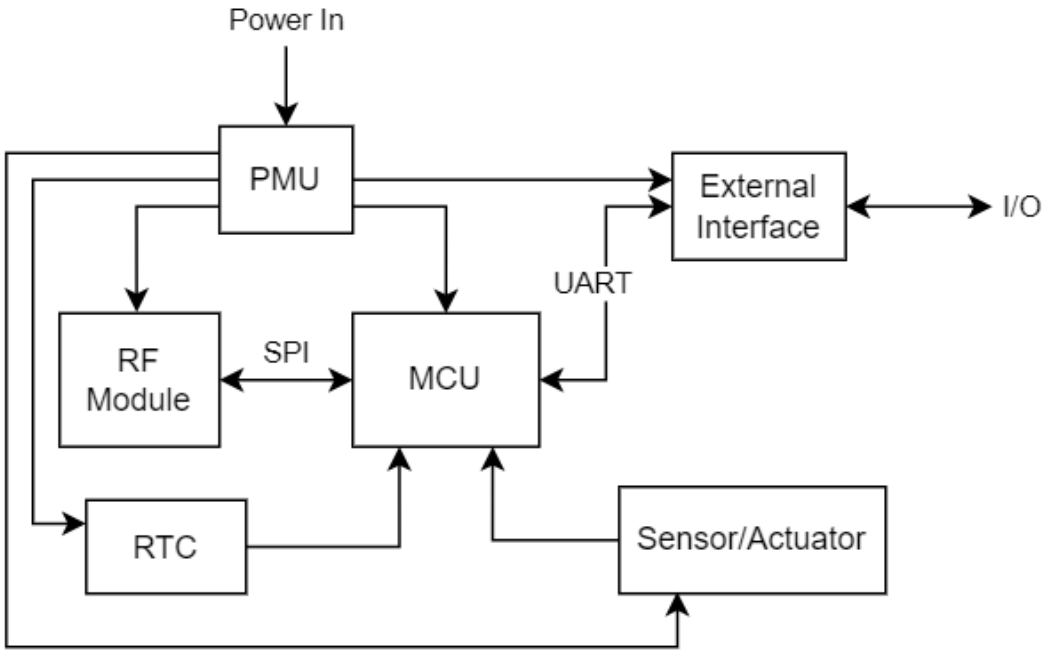

Figure 3.9: Relay Reference Design

The structure of the tunneling relay device consists of the following blocks:

- **PMU:** The power management unit is responsible for managing the power distribution, charging the battery, and providing power to the other blocks.
- **RF Module:** The RF module is responsible for channel sensing, demodulating the received signal, and modulating the transmitted signal.
- **MCU:** The microcontroller unit is responsible for managing the operation of the other blocks and running a state machine to manage the relay operation.
- **RTC:** The real-time clock is responsible for providing time information to the MCU. It is used externally to save the timestamp information.
- **External Interface:** The external interface is for providing the communication between the relay device and the other devices, programming units.

- **Sensor/Actuator:** The sensor/actuator block provides the information about the environment to the MCU and can be used to control the environment.

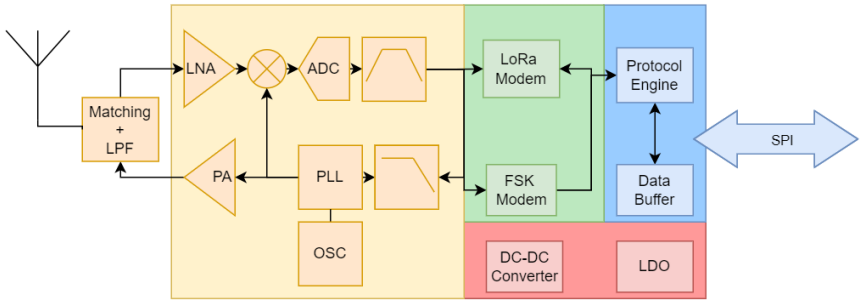

Figure 3.10: SX1262 Transceiver Block Diagram [3]

The components PMU, RTC, External Interface, Sensor/Actuator are thought as abstract blocks and can be implemented with different components. The RF Module and MCU are the main components of the relay device and should be implemented with components that are compatible with the LoRaWAN™ standard.

For the RF Module, the SX1262 transceiver[3] can be used. The block diagram of the SX1262 transceiver is given in Figure 3.10. The SX1262 transceiver is a low-power transceiver that can be used for LoRaWAN™ applications.

The MCU selection is important for the relay device since the relay device should be able to run the LoRaWAN™ stack and the relay software.

It should also support deep sleep mode to increase the battery life of the relay device. The STM32L4 series[4] can be used for the MCU selection.

The relay device state machine is given in Figure 3.11.

It is assumed that when the relay device is turned on, it is registered to the existing network and has the necessary keys to communicate with the network server. It is also assumed that the end devices that will communicate with the relay device are known by the relay device.

The relay device is in the deep sleep mode most of the time. It wakes up by a real time clock (RTC) interrupt. When the relay device wakes up, it performs the ECAD (which will be explained in Section 4.1) and checks if there is a preamble on the

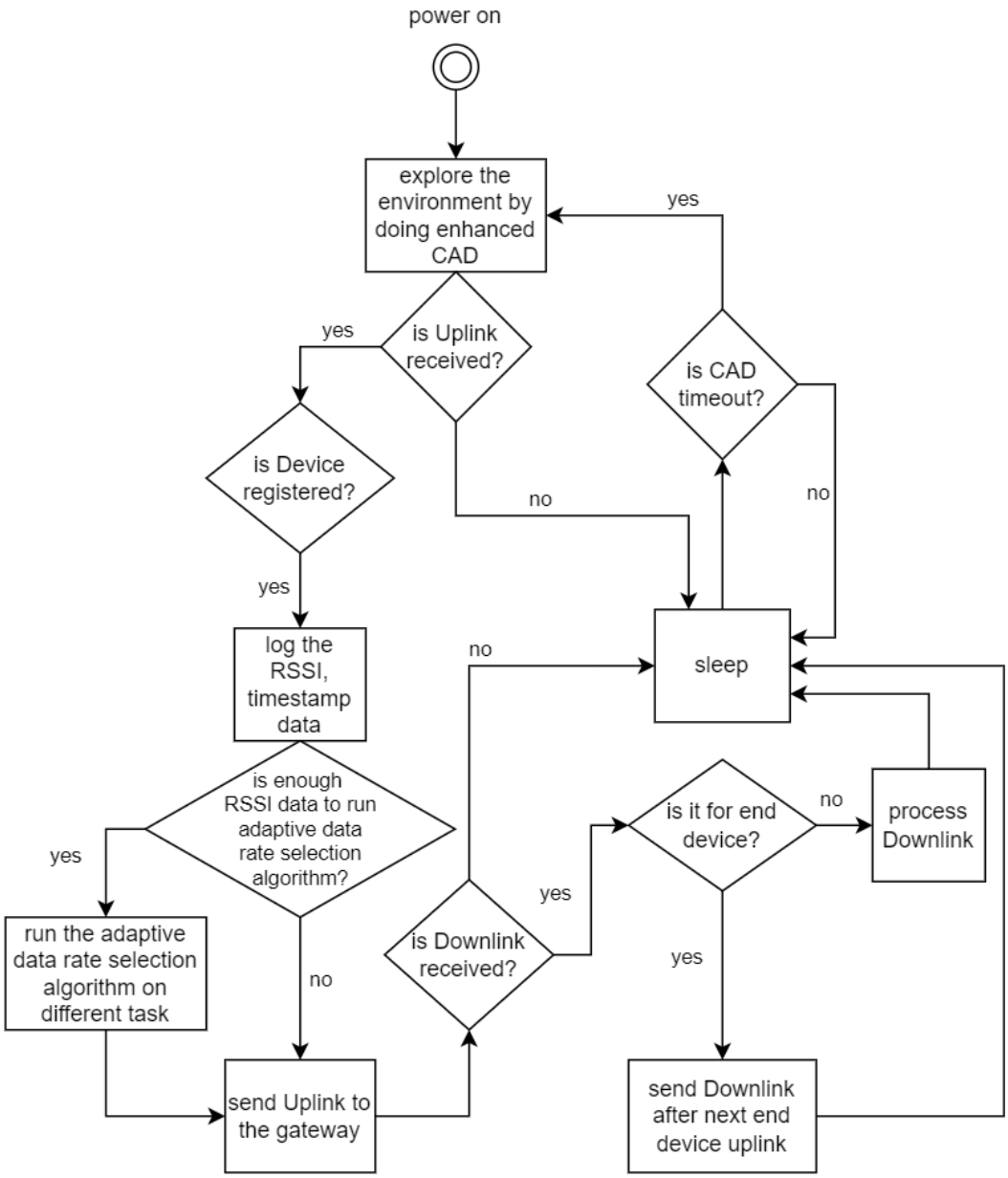

Figure 3.11: The Tunneling Relay State Machine

channel.

If a LoRa preamble is detected, the relay device tries to demodulate the received signal and checks if the received signal is a valid LoRaWAN™ frame. If the received signal is a valid LoRaWAN™ frame, the relay device checks if the received frame is valid and coming from an end device that is known by the network.

The validation is done by checking the MIC of the received frame. The relay device

sends an acknowledgment frame to the end device if the received frame is valid, then sends the received frame to the network server by encapsulating it.

3.5 System Model

Consider a closed network that has star topology and includes N end devices, M relays and 1 gateway as shown in Figure 3.12. The end devices are sending data to the gateway over the relays. The gateway is connected to a network server that is responsible for managing the network. Network server sends the data to a dashboard that is used by the user.

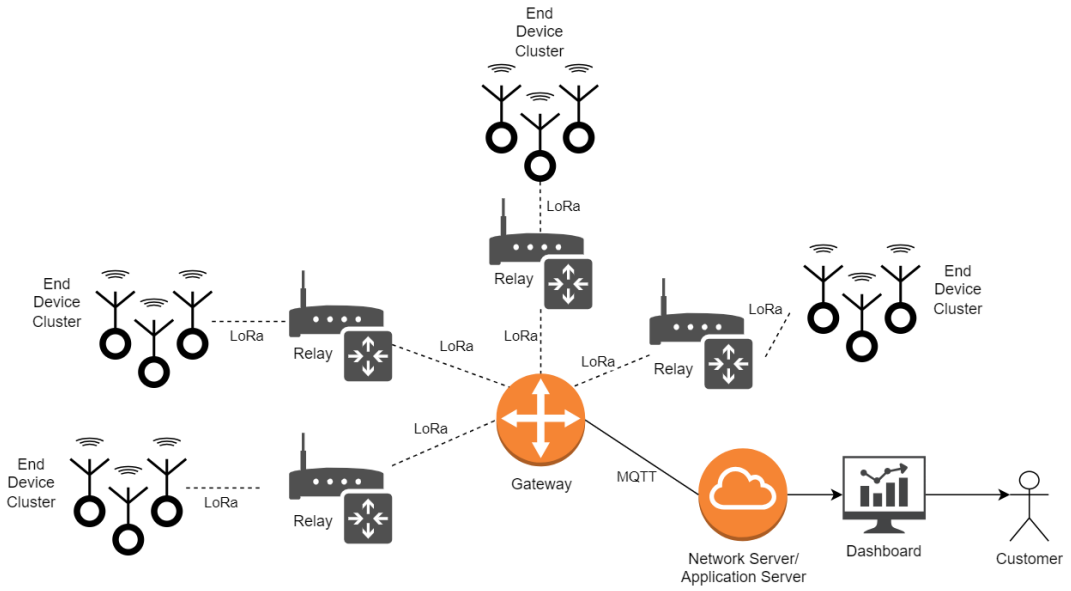

Figure 3.12: System Model for a LoRaWAN™ Network with Tunneling Relays

The end devices are sending data to the relays over the LoRa. The relays are forwarding the data to the gateway over the LoRaWAN™ protocol. The gateway is forwarding the data to the network server over the Internet, with Message Queuing Telemetry Transport (MQTT) protocol.

The end devices are using SX1262 [3] radio transceiver. The gateway is using SX1302 [45] radio transceiver. In LoRaWAN™ Relay Standard, the relay device has the same hardware capabilities as the end devices. However, in this work, it is assumed that the relay device is using SX1302 transceiver for the ease of implementation of a pro-

totype. The RF sensitivity of the SX1262 and SX1302 is given in Table 3.1. The sensitivity values are calculated with 10% packet error rate, a receiving packet of 12 Bytes. In addition to the assumptions above, SX1302 sensitivity is measured by assuming it has a SX1250 front-end module, low-noise amplifier with 18 dB gain, and 1.5 dB Noise Figure.

Table 3.1: SX1262 Receiver Sensitivity Values

@125kHz	**SX1302**	**SX1262**
SF7	-127 dBm	-124 dBm
SF10	-135.5 dBm	-
SF12	-141 dBm	-137 dBm

3.6 Packet Formats

The packet formats are distinguished by the type of links. There are two types of links in the network if the relay is on the network. The links are the end device to relay link, the relay to gateway link. In our consideration, we think of the gateway and the network server as one unit.

The packet format for the end device to relay link is given in Figure 3.13. It is a LoRaWAN™ uplink packet without any change. The packet includes header, payload, and MIC fields.

Header includes the end device identifications, service differentiation parameters and message type. The payload is the actual payload, and it is encrypted by a session key that exists in both the end device and the gateway. The MIC is the message integrity code that is calculated by using another session key and the payload.

The packet format for the relay to gateway link is given in Figure 3.14. It is a still LoRaWAN™ uplink packet without any change. The packet includes header, payload, and MIC fields. Header includes the relay identifications, service differentiation parameters and message type. The payload is the message of the end device to the gateway, and it is encrypted by a session key that exists in both the relay and the gate-

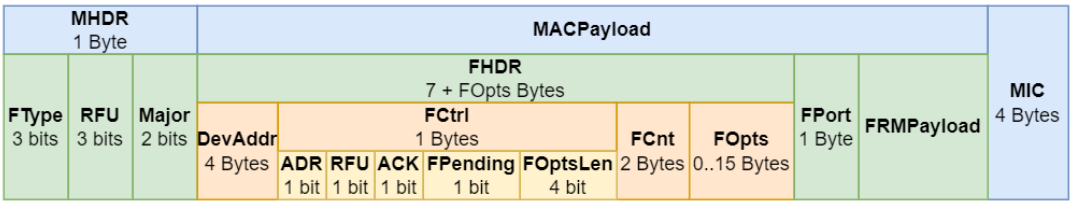

Figure 3.13: End Device to Relay Packet Format [7]

way. Relay does not know the actual content of the payload, it just forwards it to the gateway. The relay takes the uplink packet from the end device, encapsulates it with its own header, and sends it to the gateway. The MIC is the message integrity code that is calculated by using another session key and the payload.

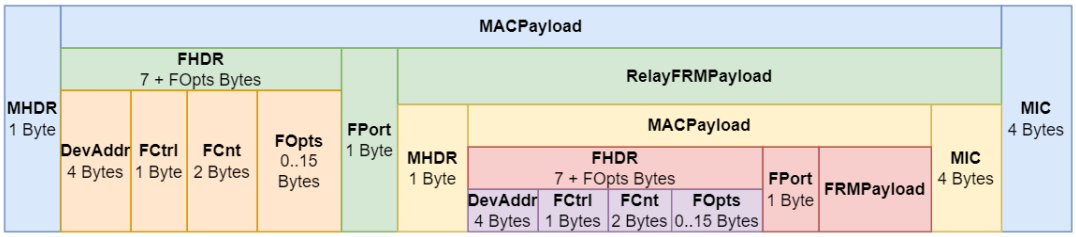

Figure 3.14: Relay to Gateway Packet Format

The downlink and uplink processes are same, just the message type is different.

There is no need to use the new packet formats defined in the LoRaWAN™ Relay Standard and given in the Figure 3.6. The relay device can use the existing packet formats and just encapsulate the uplink packets from the end devices and forward them to the gateway.

The tunneling relay comes with its new medium access control (MAC) commands.

3.6.1 The Effect of the Activation Type

The activation type of the end devices affects the relay device operation. If the end device is activated by the Over-The-Air Activation (OTAA), the relay device should be able to handle the join procedure. In the existing relay standard [8], the relay device handles the join procedure by sending related queries to the network server to

instantiate a session for the end device. Contrary to the existing relay standard, the tunneling relay, after turning on, gets the necessary prior information from the network server to communicate with the end devices by removing the overhead coming from the activation procedure. The same process is also applied for the Activation-By-Personalization (ABP) activated end devices.

Moreover, the existing relay standard requires more frames to be sent by the end device to the relay device to establish a session. The tunneling relay does not require any additional frame to be sent by the end device since the relay device already knows the end devices that will communicate with it.

3.6.2 Where Should the Calculations be Done?

The relay device should be able to calculate the needed parameters to forward the uplink packets from the end devices to the gateway. In the existings relay standard, the relay device is just responsible for forwarding the uplink packets from the end devices to the gateway. However, if we consider the edge computing concept, the relay device can be used to calculate some parameters to reduce the power consumption of the end devices. The main consideration is to reduce the power consumption of the end devices. Moreover, we should decide whether the relay device be a part of the network server, like a proxy, since it has the ability to recognize end devices, or will it be an independent device that just forwards the packets. The system model for the relay device can be extended to include the edge computing concept.

3.6.3 New MAC Commands

The tunneling relay introduces new MAC commands to the network. The new MAC commands are used to manage the relay device behavior and the network behavior towards the relay device. The new MAC commands are given in Table 3.2.

The new MAC commands are considered as an extension to the existing relay MAC commands. The MAC commands are sent in the same way as the existing relay MAC commands. The network server sends the MAC commands to the relay device by

Table 3.2: New MAC Commands for Tunneling Relay

CID	MAC Command	Transmitted By		Description
		Relay	Network-Server	
0x49	***RelayFlushDevicesReq***		✓	Request from the network server to the relay device to flush the end device list.
0x4A	***RelayFlushDevicesAns***	✓		Answer from the relay device to the network server to confirm the end device list is flushed.
0x4B	***RelayDeviceStatusReq***		✓	Request from the network server to the relay device to get the status of an end device.
0x4C	***RelayDeviceStatusAns***	✓		Answer from the relay device to the network server to give the status of an end device.

using the gateway. The relay device sends the MAC commands to the network server by using the gateway. An example of a MAC command packet format is given in Figure 3.15.

3.6.3.1 RelayFlushDevicesReq

The **RelayFlushDevicesReq** MAC command is used by the network server to request the relay device to flush the end device list.

The end device list is the list of end devices that are known by the relay device. The

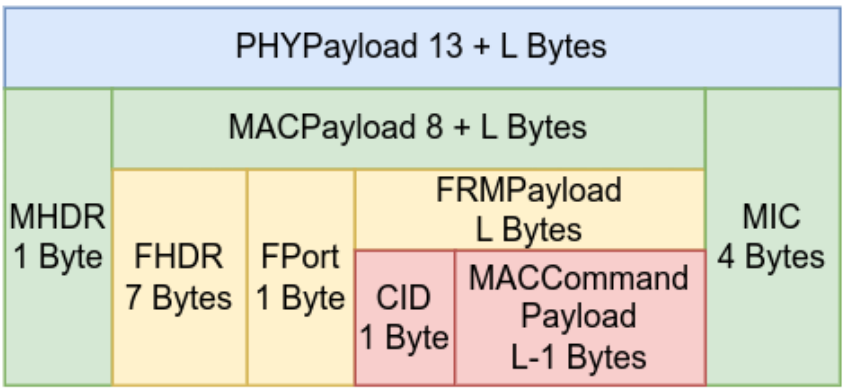

Figure 3.15: MAC Command Packet Format

relay device should flush the end device list if the network server wants to remove an end device from the relay device. This MAC command does not include any payload, it is just a request with the command identifier (CID) to the relay device.

The relay device should send a **RelayFlushDevicesAns** MAC command to the network server to confirm that the end device list is flushed.

3.6.3.2 RelayFlushDevicesAns

The **RelayFlushDevicesAns** MAC command is used by the relay device to confirm the network server that the end device list is flushed. The relay device should send this MAC command to the network server after receiving the **RelayFlushDevicesReq** MAC command.

The MAC command includes the command identifier (CID) and the status of the operation as 1 Byte. The status field is shown in Table 3.3.

Table 3.3: Status Field for ***RelayFlushDevicesAns*** MAC Command

Bits	7:2	1:0
Field	RFU	Status

The status of the operation can be success or failure (1 or 0).

3.6.3.3 RelayDeviceStatusReq

The **RelayDeviceStatusReq** MAC command is used by the network server to request the relay device to get the status of an end device. The network server should send this MAC command to the relay device to get the status of an end device.

The MAC command includes the command identifier (CID) and the device unique identifier (DevEUI), which is 8 Bytes of the end device that the network server wants to get the status.

The relay device should send a **RelayDeviceStatusAns** MAC command to the network server to give the status of the end device.

3.6.3.4 RelayDeviceStatusAns

The **RelayDeviceStatusAns** MAC command is used by the relay device to give the status of an end device to the network server. The relay device should send this MAC command to the network server after receiving the **RelayDeviceStatusReq** MAC command.

The MAC command includes the command identifier (CID), the device unique identifier (DevEUI), which is 8 Bytes of the end device that the network server wants to get the status, the metadata of the end device, as 3 Bytes, and the uplink frequency of the end device.

The metadata of the end device is shown in Table 3.4.

Table 3.4: Metadata Field for ***RelayDeviceStatusAns*** MAC Command

Bits	23:13	12:11	10:4	3:0
Field	RFU	Battery Level	Uplink RSSI	Uplink SNR

The battery level is the battery level of the end device, the uplink RSSI is the received signal strength indicator of the uplink frame from the end device, and the uplink SNR is the signal-to-noise ratio of the uplink frame from the end device.

3.7 Link Budget Calculation

Consider the Friis transmission equation[46]:

$$\frac{P_r}{P_t} = \left(\frac{A_r A_t}{d^2 \lambda^2}\right) \tag{3.4}$$

Where P_t is the total signal power at the transmitting antenna output terminals, P_r is the power at the receiving antenna, A_r is the effective aperture area of the receiving antenna, and A_t is the effective aperture area of the transmitting antenna, d is the distance between antennas, λ is the wavelength of the radio frequency.

If we inspect the effective aperture area of an antenna as gain and directivity metrics, we obtain:

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi d}\right)^2 \tag{3.5}$$

Where G_t and G_r are the antenna gains of the transmitting and receiving antennas with respect to the isotropic radiator model.

$$P_r{}^{[dB]} = P_t{}^{[dB]} + G_t{}^{[dBi]} + G_r{}^{[dBi]} + 20\log_{10}\left(\frac{\lambda}{4\pi d}\right) \tag{3.6}$$

If we extend the last argument of the Equation (3.6) as free space path loss (FSPL), we get:

$$P_r = P_t + G_t + G_r - 20\left(\log_{10}(d) + \log_{10}(f) + \log_{10}\left(\frac{4\pi}{c}\right)\right) \tag{3.7}$$

If we consider the frequency is in MHz and the distance is in kilometers, the FSPL equation becomes:

$$FSPL = L_{PL} = 32.45 + 20\log_{10}(f) + 20\log_{10}(d) \tag{3.8}$$

For dB scale, when the other losses are added, the received power P_r is calculated in Equation (3.9).

$$P_r = P_t + G_t - L_T - L_{PL} + G_r - L_R \tag{3.9}$$

Where P_t is the transmission power, G_t is the gain of the transmitting antenna, L_T is the transmission line losses, L_{PL} is the free space path loss, G_r is the gain of the receiving antenna, and L_R is the receiving line losses. It is assumed that, the network constellation is on $EU868$ Region in RP002[6], and all the gateway, the relay, and the end devices use the same omnidirectional 868 MHz antenna with a gain of 3 dBi (which is a common antenna that can be easily found on market). Reference link budget diagram is given in Figure 3.16.

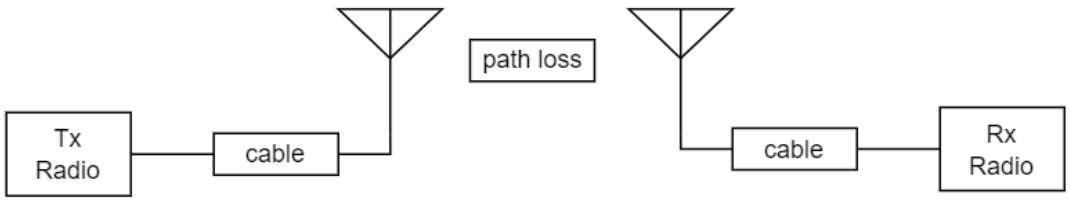

Figure 3.16: Reference link budget diagram

For EU868 region defined in [6], the maximum transmission power is used in the link budget calculation which is 14 dBm [6][47]. The cable losses at the receiver and the transmitter part is taken as 1 dB for transmission and reception circuits. The antenna gains are taken as identical and equal to 3 dBi. With these assumptions, the maximum distances for different spreading factors is calculated and given in Table 3.5.

However, the FSPL is not the only environmental factor that effects the link budget. A real experimental setup is required to model the exact communication channel and find the link budget. In [2], the link budget is calculated for a real experimental setup in an urban area, Shinjuku, Japan and the results are given in Table 3.6 and the plan of the urban area is given in Figure 3.17.

Table 3.5: Maximum distances for different spreading factors with SX1302 and SX1262

@868MHz	**SX1302**	**SX1262**
SF7	60 km	40 km
SF10	150 km	-
SF12	300 km	190 km

Table 3.6: Shinjuku Urban Range Test [2]

Ref.	Distance (m)	LoRa:125 kHz B, SF = 8 (3.125 kb/s)	
		RSSI (dBm)	SNR (dB)
A	80	-91	0
B	150	-102	0
C	280	-114	0
D	330	-124	10
E	480	-120	0
F	560	-121	0
G	1180	-112	0
H	1350	-126	10
I	1750	-127	100

3.8 Capacity Analysis of a Single Channel Relay

To find the capacity, first, we need to find the corresponding data rates for each SF value. Shannon's theorem[48] can be considered in Equation (3.10).

$$C = B \times \log_2 \left(1 + \frac{S}{N}\right) \tag{3.10}$$

Where C is the channel capacity, B is the channel bandwidth in Hz, S is the average received signal power in Watts, N is the average noise or interference power in Watts, S/N is the average SNR. The energy for a single bit is written as E_b, and it is equal

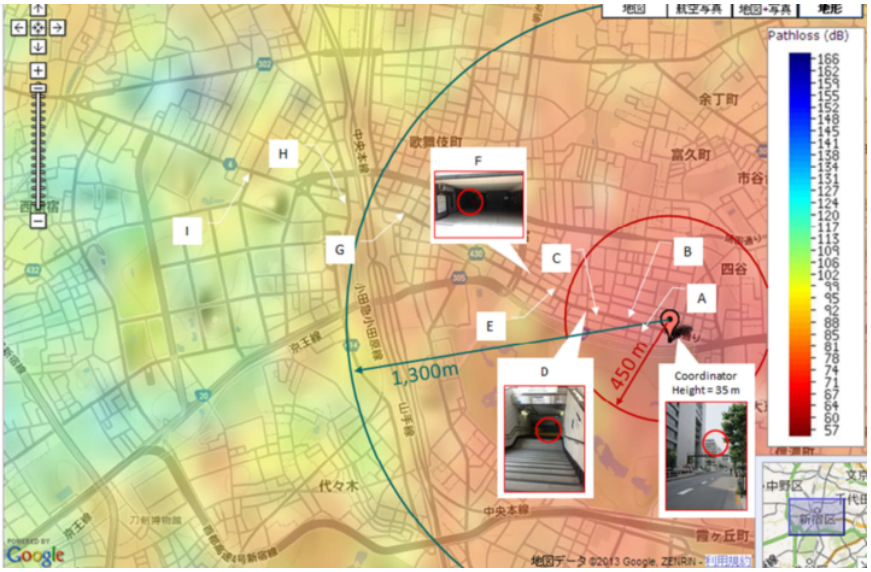

Figure 3.17: Shinjuku Urban Area [2]

to the multiplication of bit time T_b and the signal power S. The noise power is N, and it is calculated by the multiplication of noise power spectral density N_0 and the bandwidth B.

From [2], the bit rate R_b is upper bounded by the capacity.

Then, we can write the capacity as:

$$\frac{C}{B} = \log_2\left(1 + \frac{E_b}{N_0}\frac{C}{B}\right) \tag{3.11}$$

Therefore, we can express the required SNR as:

$$\frac{S}{N} = \frac{E_b}{N_0}\frac{C}{B} \tag{3.12}$$

Since the right-hand side is generally small for spread spectrum applications, and E_b/N_0 is fixed, which means for all data rates, bit energy is fixed for LoRa, the equation can be simplified as [49]:

$$\frac{C}{B} = 1.433 \times \frac{S}{N} \tag{3.13}$$

It can be seen that, in a fixed SNR channel, the transmitted signal bandwidth needs to be increased to reduce the error rate.

For direct sequence spreading, the amount of spreading is depending on the "chips per bit" ratio of the chip sequence to the desired data rate, which is referred as processing gain G_p, expressed in dB.

$$G_p = 10 \times \log_{10}\left(\frac{R_c}{R_b}\right)(dB) \tag{3.14}$$

Where R_c is chip rate per second, and R_b is bit-rate per second.

The relation between data bit rate, symbol rate and chip rate for LoRa is as follows:

$$R_b = SF \times \frac{1}{\left(\frac{2^{SF}}{B}\right)} \tag{3.15}$$

Where SF is the spreading factor, varying from 7 to 12 and B is the bandwidth.

The symbol period is T_s defined as,

$$T_s = \frac{2^{SF}}{B} secs \tag{3.16}$$

And the symbol rate is,

$$R_s = \frac{1}{T_s} = \frac{B}{2^{SF}} symbols/sec \tag{3.17}$$

The chip rate is,

$$R_c = R_s \times 2^{SF} chips/sec \tag{3.18}$$

Since one chip is sent per second per Hz of bandwidth,

$$R_c = R_s \times 2^{SF} R_c = \frac{B}{2^{SF}} \times 2^{SF} chips/sec \tag{3.19}$$

In LoRa, error correction is also used. Thus, the nominal bit rate of the data signal is:

$$R_b = SF \times \frac{CR}{\left(\frac{2^{SF}}{B}\right)} \tag{3.20}$$

Where CR is the code rate, and generally it is $4/5$.

For this scenario, it is assumed that, preamble with 12 symbol length is used, header is assumed as 8 bytes and 4 bytes MIC is also added to the end of the PHYPayload, the payload size is assumed as 30 bytes.

Therefore, bit rates and on air times for $4/5$ code rate and for 125 kHz bandwidth is given in Table 3.7.

Table 3.7: Bit Rates for Different Spreading Factors

SF	**Bit Rate (bps)**	**Time on Air (ToA) (ms)**
7	5468	87.3
8	3125	154.1
9	1758	287.7
10	977	534.5
11	537	1151
12	293	2138

In [2], the capacity of the LoRa modulation C_{LoRa} is calculated for spreading factors (SF), for 6 to 12, and found as 21.531 kbps. The Shannon capacity, with the same N_0 value, which is $-53.63dBm/Hz$, with $125kHz$ B, and $14dBm$ transmission power, is calculated in Equation (3.21). The N_0 value, as stated in [50], close to the urban area noise levels.

$$\begin{aligned} SNR &= \tfrac{P_S}{P_N} = 92.85W \\ C &= B \times \log_2(1 + SNR) \\ C &= 125000Hz \times \log_2(1 + 92.85) \\ C &\approx 817.5kbps \end{aligned} \tag{3.21}$$

It is observable that the Shannon capacity C upper bounds the LoRa capacity C_{LoRa} for a single channel with $125kHz$ bandwidth. However, there is a huge gap between two capacities. The intuition behind this is, the Shannon capacity assumes the ideal conditions such as, perfect channel conditions, optimal modulation schemes. Moreover, the design goal of LoRa modulation is transmitting low-sized payloads (less than 100 Bytes typically), over long ranges by also considering the power efficiency. Therefore, the practical data rates achievable by LoRa are much lower due to its design optimizations for range, power efficiency, and robustness in noisy environments. This difference is expected and reflect the trade-offs inherent in the design of different communication systems.

CHAPTER 4

ENHANCEMENTS IN TUNNELING RELAY

4.1 Enhanced Channel Activity Detection (ECAD)

Channel Activity Detection (CAD) is a process that is used for detecting the presence of a signal on a single channel. In LoRa, the CAD is used to detect the preamble of a LoRa frame. For the relay device, the CAD is used to detect a WOR frame on a single channel.

The relay device wakes up periodically to perform the CAD operation. In LoRaWAN™ Relay Specification [8], the default CAD period is 1000 milliseconds. There are two relay channels, one is mandatory, and the other one is optional.

On the Relay Specification defined in [8], the end device must send a WOR frame on the mandatory relay channel to reach the gateway over the relay. The length of a payload of WOR frame is 15 Bytes as stated in Figure 3.3.

It is suggested that the end device should send the WOR Uplink frame with a long preamble to increase the probability of detection by the relay device.

The ECAD is a process that is used to detect the uplink frames from the end devices instead of the WOR frames. The relay device wakes up periodically to perform the ECAD operation.

The aim is to detect the uplink frames directly from the end devices with a low power consumption. To do that, it is better to provide a new system model for the ECAD operation.

4.1.1 System Model for the Performance Study of ECAD

Imagine a relay device that is outfitted with a singular antenna, designed to handle both sending and receiving signals. To simplify the conceptual model, it's posited that this relay device is tuned to receive uplink transmissions exclusively from one channel. This device is programmed to intermittently wake from a dormant state to execute the ECAD procedure.

For our simulation scenario, there exist N distinct end devices, which are engaged in dispatching uplink transmissions towards the channel monitored by the relay device. These N devices are positioned equidistantly from the relay device, and they both broadcast their signals with identical power levels and utilize the same spreading factor for their transmissions for the sake of simplicity of the simulation environment.

Additionally, the data payload contained within the uplink transmissions for all devices is of equal size. These end devices are scheduled to send their uplink transmissions in a periodic manner, ensuring that their signals do not interfere with each other.

However, a complication arises when there are unregistered or unauthorized end devices sending data to the same communication channel that relay listens to. Since the frequency band is shared by public and private networks, the relay device should be able to distinguish the uplink frames from the legitimate end devices and the uplink frames from the unauthorized end devices. Since the relay does not know which device is transmitting without demodulating the received signal after CAD process, demodulating the packets coming from unknown devices will cause false alarm. The false alarm happens when the relay demodulates packets that are not relevant to the network.

The challenge for the relay device lies in its ability to accurately identify and process uplink frames originating from legitimate end devices, achieving this with a minimal rate of false alarm while maintaining a high success rate in detection. The end device is defined as legitimate if it was activated by the network. The activation types are defined in Chapter 2 and in [7].

The illustration of system model is given in Figure 4.1.

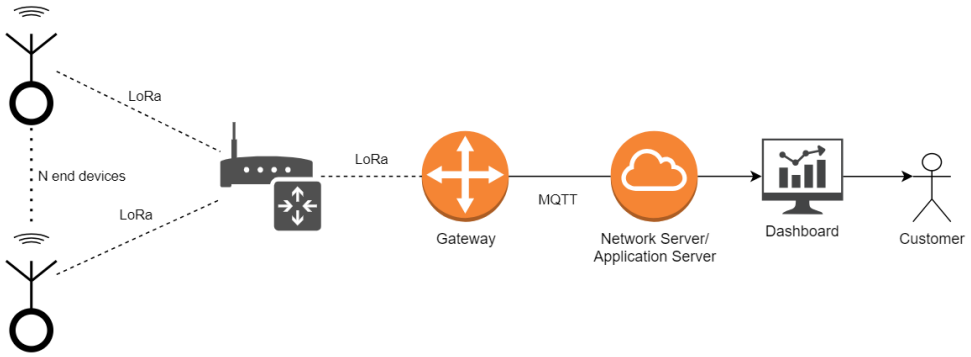

Figure 4.1: ECAD System Model

To make the relay device detect the uplink frames from the legitimate end device with a low probability of false alarm and a high probability of detection, the relay device should implement a detection algorithm. The detection algorithm should adjust the channel scan times.

The end device credentials such as session keys or network keys (depending on the activation type is over-the-air activation (OTAA) or activation by personalization (ABP)), is given to the relay device after deployment as stated in the current specification[8]. The relay device communicates with the gateway to get that credentials by MAC command interface defined in [7].

In LoRaWAN L2 Layer Specification [7], there are MAC commands that are defined already to maintain and configure the operation of the end devices and the relay devices. In relay specification [8], as stated in the flow in Figure 3.7, the relay device does not know about the end devices until the end devices send a WOR frame to the relay device.

These enhancements and system model defined here can be thought as an extension to the relay specification.

If the relay device knows the end devices that are registered to the network, at least the end devices that will reach the gateway via this relay device, the overhead of the relay-gateway communication can be decreased. Because in the existing standard, the relay device should communicate with the gateway to get the end device credentials

to demodulate the uplink frames coming from the end devices, for each first contact with the end device. We can consider the relay device as a proxy of the network server for the end devices that are already registered to the network. As the relay device knows which end devices will reach the gateway via the relay itself, the relay can explore the end devices that are registered to the network by listening the channel and estimating the uplink frame arrival times of the end devices.

This estimation algorithm can be formulated as a Multi Armed Bandit (MAB) problem.

4.1.2 Problem Definition

The relay device periodically wakes up to listen to the channel to detect the uplink frames from the end devices. The Channel Activity Detection (CAD) activity takes a few milliseconds depending on the spreading factor [51].

However, the periodicity of the Channel Activity Detection (CAD) process is unknown at the beginning. The aim of the ECAD algorithm is to converge channel scan times that match the end devices' uplink frame arrival times.

The uplink frame arrival times are assumed to be periodic for each device in the network. Since the relay is deployed to a network that has already end devices and a gateway, it is also assumed that the relay device have a prior knowledge about the end devices' uplink frame periodicities. Let the uplink frame periodic arrival offsets of a set of N end devices be $\{\tau_1, \tau_2, \ldots, \tau_N\}$. The number N is a multiplier of the number of end devices at the related field and it is also a multiplier of the periodicity of the uplink frame arrival times of the end devices.

The relay device should listen to the channel at the time that the uplink frame is expected to arrive. This means the relay device should estimate the time offset of the uplink frame arrival times of the end devices.

The CAD duration is varying for different SFs, similar to the relay specification [8], for Wake-on-radio (WOR) frames the spreading factor is set to 9, it is assumed that the end devices are using the same SF and the periodicity of the end devices uplink frame

arrival times are not colliding with each other. Since the aim of the CAD process is to detect a LoRa preamble, it should listen at least L_{nbS} symbols to detect the preamble, which is varying for different SFs, and L_{nbS} is the minimum number of symbols that should be captured to detect the preamble, varying for different SFs.

Let the preamble duration be T_{preamble} and the preamble length be L_{preamble} symbols. To decrease the false alarm rate, which is, detecting a LoRa transmission when there is none or from a device that does not belong to the network, it is assumed that the relay device should listen the whole preamble to detect the uplink frame.

If the time is divided to slots with a duration of $\tau_n / L_{\text{preamble}}$, the relay device should listen to the channel at the slot that the uplink frame is expected to arrive. The slots represent the preambles of the uplink frames coming from the end devices. The slots are thought as the arms of a Multi Armed Bandit (MAB) problem.

Let the action set, which contains arrival slot indices of devices be $\mathcal{A} \in \{1, 2, \ldots, N\}$ where N is the number of slots in a prior period. The relay device should select an action α_i at each time step to maximize the reward function.

The action α_i is the decision of the relay device to listen to the channel at the slot i.

For an action $alpha_i \in \mathcal{A}$ where $i \in 1, 2, \ldots, N$, we define a reward function $r_i(t)$ for the action α_i at time t is defined in Equation (4.1):

$$r_i(t) = W_i \left[\cos\left(\tfrac{2\pi}{\tau} t\right) \cos\left(\Delta\phi_i\right) - \sin\left(\tfrac{2\pi}{\tau} t\right) \sin\left(\Delta\phi_i\right) \right] \tag{4.1}$$

Where $\Delta\phi_i$ is the phase shift of the action α_i and it is equally likely to take each value in $\{\frac{2\pi k}{N}, k = 0, 1, \ldots, N-1\}$. The weight W_i is proportional to Received Signal Strength Indicator (RSSI) of the ith arm, which is directly related with the distance between the end device and the relay device. W_i is the weight of the action α_i and $W_i = cRSSI_i$ where c is a constant, and $r_i(t)$ is the reward of the action α_i at time t. The reward function is a periodic function that is varying for different arms with time. Let the received signal strength indicator (RSSI) of the uplink frame be $RSSI_i$ for the action α_i.

We will develop a policy π that select the action $\alpha_i(t)$ at time t to maximize the

mean expected reward functions, by providing a heuristic solution to MAB problem to estimate the phase of the reward functions. This work considers the end devices that are transmitting periodical uplink frames to the network. We can define the reward maximization problem as sensing the time slot that gives the maximum reward by learning its time offset as the time goes on, given in Equation (4.2).

The reward function indicates that a relay device should listen to the channel at the exact time instance which there is a packet on the channel coming from the desired end devices. The more lower the reward value means the more closeness to the packet coming after the preamble. The relay device can be sure that if it receives the exact number of symbols that an uplink packet has, it can be sure that the timing is correct for the uplink frame.

$$\max_{\pi \in \Pi} \quad \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[r(t)] \tag{4.2}$$

Consider policy π_* that maximizes the expected reward function by selecting the best action at each time step. The expected regret of the policy π is defined in Equation (4.3).

$$\mathbb{E}[R_T(\pi)] = \frac{1}{T} \sum_{t=1}^{T} \big(\mathbb{E}[r^*(t)] - \mathbb{E}[r(t)] \big) \tag{4.3}$$

The instantaneous regret of the policy π is defined in Equation (4.4).

$$R(t) = r^*(t) - r(t) \tag{4.4}$$

Where $\mathbb{E}[R_T(\pi)]$ is the expected regret of the policy π, $r^*(t)$ is the reward of the best action at time t, and $r(t)$ is the reward of the action α_i at time t.

There exists different approaches to solve a MAB problem with a periodic reward function.

In [52], the authors modeled the reward function as a periodic Gaussian process which is a non-stationary stochastic process. The proposed Periodic-GP algorithm uses pe-

riodic kernel function to model the periodic reward function, with a known prior period.

In [53], GP-UCB algorithm is proposed to solve the Multi Armed Bandit (MAB) problem by using Reproducing Kernel Hilbert Space (RKHS) to model the reward function.

In [54], time is divided into smaller intervals and piece-wise stationary reward function is modeled.

All these algorithms can be used to solve the ECAD problem. For the relay case, it can be assumed that the relay has a prior knowledge about the end devices uplink frame periods. Moreover, the algorithms defined above are not suitable for the ECAD problem since the relay device is battery powered and the algorithms require a unnegligible processing power.

4.1.3 ECAD Algorithm

Below, the key steps and mathematical formulation of the ECAD algorithm are presented.

The pseudocode of the ECAD algorithm is given in Algorithm 4.

4.1.3.1 Initialization Phase

The algorithm starts with an initialization phase where the expected reward, denoted as Q_i for arm i, and corresponding number of times the arm is selected, denoted as N_i, are initialized to zero for all arms.

For each arm, we will define coefficients, denoted as a_0, a_1, and b_1, which are also initialized to zero. The coefficients are used to model the reward function of an arm. A data structure, stored locally, is initialized to store the rewards and the times the arms are selected. Then, the algorithm proceeds with an approximation of the reward function of each arm.

Algorithm 4 ECAD Algorithm

Initialization:

Set $Q_i = 0$ and $N_i = 0 \; \forall i \in 1, 2, \ldots, N$

Set $a_0 = 0$, $a_1 = 0$, and $b_1 = 0 \; \forall i \in 1, 2, \ldots, N$

Initialize the data structure to store the rewards and the time slots the arms are selected

for $t = 1, 2, \ldots, T$ **do**

Calculate $Q_i(t)$ using Equation (4.5) $\forall i \in 1, 2, \ldots, N$

Select $i = \arg\max_i Q_i(t)$

Update $N_i(t) = N_i(t) + 1$

Receive the actual reward $r_i(t)$

Update $Q_i(t)$ using Equation (4.6)

Update a_0, a_1, and b_1 using Equation (4.7a)

end for

4.1.3.2 Selecting An Arm

At each time step, the algorithm selects an arm to play based on the estimated reward of each arm. The estimated reward of an arm is calculated by using the coefficients of the reward function of the arm, the calculation is given in the Equation (4.5).

$$\hat{Q}_i = \sum_{t=0}^{T} \left(a_0 + a_1 \cos\left(\frac{2\pi t}{\tau} + \Delta\phi_i\right) + b_1 \sin\left(\frac{2\pi t}{\tau} + \Delta\phi_i\right) \right) \tag{4.5}$$

Where $\hat{Q}_i$ is the estimated reward of the arm i, where $i = 1, 2, \ldots, N$, a_0, a_1, and b_1 are the coefficients of the reward function of the arm i and set to zero at $t = 0$, and t is the current time. The algorithm selects the arm that has the maximum estimated reward. We can truncate the sum to a finite number of terms to calculate the estimated reward.

4.1.3.3 Updating the Reward by Experience

After selecting an arm, the algorithm receives the actual reward for an arm, and updates the expected reward for an arm as given in Equation (4.6).

$$Q_i(t) = Q_i(t-1) + \frac{1}{N_i}\left(r_i(t) - Q_i(t-1)\right) \tag{4.6}$$

Where $Q_i(t)$ is the expected reward of the arm i at time t, N_i is the number of selection times of arm i, and $r_i(t)$ is the actual reward of the arm i at time t.

4.1.3.4 Updating the Coefficients

The coefficients of the reward function of an arm are updated by using the actual rewards of the arm.

$$a_0 = \frac{2}{T}\sum_{t=1}^{T} r_i(t) \tag{4.7a}$$

$$a_1 = \frac{2}{T}\sum_{t=1}^{T} r_i(t)\cos\left(\frac{2\pi t}{\tau}\right) \tag{4.7b}$$

$$b_1 = \frac{2}{T}\sum_{t=1}^{T} r_i(t)\sin\left(\frac{2\pi t}{\tau}\right) \tag{4.7c}$$

Where a_0, a_1, and b_1 are the coefficients of the reward function of the arm i, T is the total number of time steps, and $r_i(t)$ is the actual reward of the arm i at time t.

CHAPTER 5

RESULTS AND DISCUSSIONS

This chapter presents the results of the simulation study conducted to evaluate the performance of the proposed enhancements to the relay device and the network. The simulation study focuses on the ECAD mechanisms. The results are discussed in detail, highlighting the impact of the proposed enhancement on the network performance. Additionally, the chapter includes a detailed description of the LoRa end device PCB design and the patent application for the proposed relay device.

5.1 ECAD Simulation Results

In this section, the simulation results of the ECAD mechanism are presented. The ECAD algorithm is compared with the Upper Confidence Bound (UCB) algorithm, Greedy algorithm, ϵ-Greedy algorithm, and TS algorithm. The simulation study evaluates the performance of the proposed ECAD mechanism in terms of the expected reward, instantaneous regret, and cumulative regret. The impact of the number of fraud end devices on the total reward and total regret is also analyzed. The fraud device is defined as the device that sends uplink data but did not register to the network.

5.1.1 Simulation Parameters

The simulation parameters used for the expected reward comparison, instantaneous regret comparison, and cumulative regret comparison are stated in Table 5.1.

In 30 time slots, there is only one time slot that the optimal end device is transmitting. The aim of the relay is to converge to that time slot to maximize its reward (in our

Table 5.1: Simulation Parameters for Expected Reward, Instantaneous Regret, and Cumulative Regret Comparison

Parameter	**Value**
Number of slots	30
Optimal Arm Index	15
Number of Empty Scans	15
Number of Frauds	14
Number of Rounds	100
Number of Simulations	100
Periodicity	30
Q1	1
ϵ	0.1

case it is the low power consumption). The optimal arm index, the actual end device to listen, is selected as in mod of 15th slot to make the simulation more challenging. The empty scan is defined as running CAD process without detection of any LoRa signal, which costs as the power consumption for CAD process only. The number of rounds is set to 100, and the number of simulations is set to 100 to obtain reliable results. The periodicity is set to 30 to make the simulation more realistic. The $Q1$ value is the starting value of the reward function for the Greedy algorithm, and the ϵ value is the exploration rate for the ϵ-Greedy algorithm. The fraud device defined as the device that sends uplink data but did not register to the network, and the number of fraud devices is set to 14. Since the periodicity is 30, the relay has 30 time slots to listen for the desired end device.

The algorithms are compared based on total reward, total regret on increasing the number of fraud end devices for 100 runs. The total reward is calculated as the sum of the rewards obtained from each end device, while the total regret is calculated as the difference between the maximum reward and the total reward obtained from each end device. Moreover, the power consumption of the ECAD mechanism is compared with the CAD mechanism in LoRaWAN™ networks.

5.1.2 Reward Function Used in The Simulation

Examining the rewards allocated to each end device, as illustrated in Figure 5.1, reveals the presence of one optimal arm and one fraudulent arm. The remaining arms correspond to empty scans accompanied by noise. The optimal arm is characterized by yielding the highest reward, indicating the most favorable outcome. Conversely, the fraudulent arm is identified by producing the lowest reward, indicating the least favorable outcome. Because, the CAD and demodulation processes are run, but the signal is not the one that is expected, which results a negative reward that is lower than the empty scan reward.

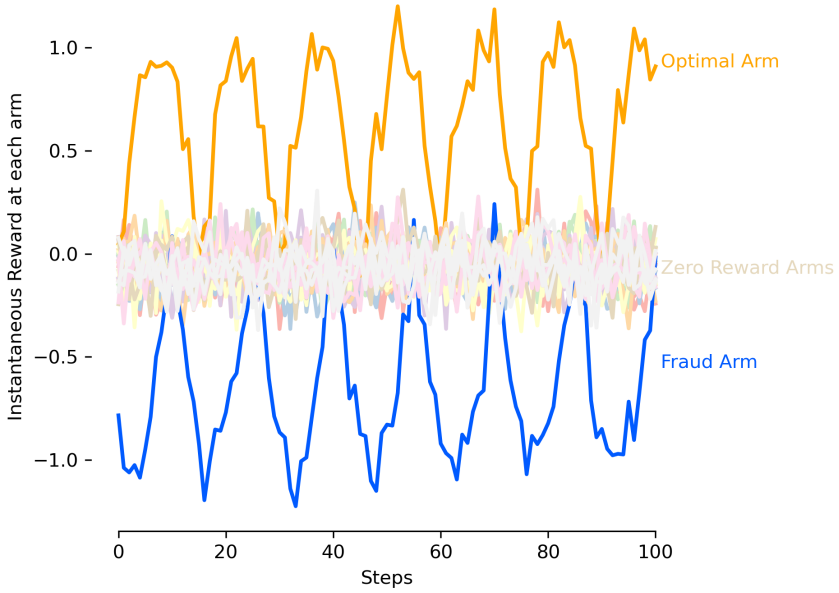

Figure 5.1: Rewards of the End Devices

5.1.3 Expected Reward Comparison

In this subsection, the expected reward comparison of the ECAD mechanism with the UCB, Greedy, ϵ-Greedy, and Thompson Sampling algorithms is presented. The comparison is made based on the average expected reward obtained from 100 simulations. The optimal policy is the one that only performs CAD operation when the one end device that is registered to the network is transmitting, then demodulates the

signal. Therefore, the only power consumption for the optimal policy occurs when it listens and demodulates the registered end device. The expected reward comparison is shown in Figure 5.2.

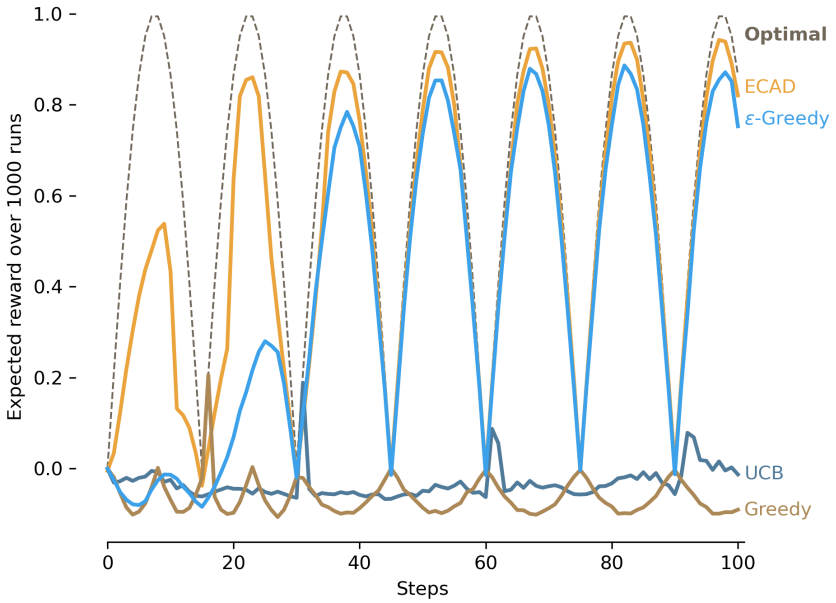

Figure 5.2: Expected Reward Comparison

The ECAD mechanism outperforms the UCB, Greedy, ϵ-Greedy, and Thompson Sampling algorithms in terms of expected reward. It converges to the optimal arm faster than the ϵ-Greedy algorithm, showcasing its effectiveness in efficiently identifying and exploiting the most rewarding arms in the multi-armed bandit problem. Additionally, during the second period of the reward, the ECAD algorithm surpasses the ϵ-Greedy algorithm by swiftly discovering the optimal arm. This further underscores the ECAD algorithm's efficiency and effectiveness in maximizing expected reward.

The ϵ-Greedy algorithm exhibits a comparable performance to the ECAD mechanism, albeit with a slight delay in converging to the optimal arm. It is better to observe the performances of the ϵ-Greedy algorithm and the ECAD algorithm with increasing the number of end devices. The comparison is made based on the total reward obtained from 1000 simulations, as shown in Figure 5.3. In simulation, there is one end device that is transmitting uplink data periodically, and the relay should demodulate the sig-

nal to get the reward. If the relay listens wrong time slot, it will get a noise signal, and the reward will be close to zero.

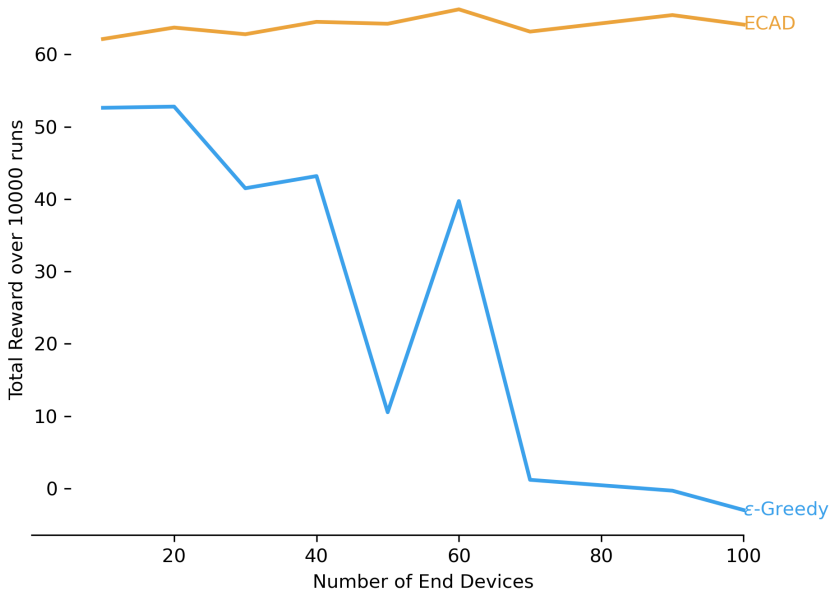

Figure 5.3: Total Reward versus Number of End Devices

As the number of end devices increases, the performance of the ECAD mechanism remains consistent, demonstrating its robustness and adaptability in dynamic environments. Contrary to ECAD, the ϵ-Greedy algorithm exhibits an unstable behavior in performance as the number of end devices increases. The main reason behind the performance degredation of the ϵ-Greedy algorithm is its static exploration rate, which does not adapt to the changing environment.

The total regret comparison of the ECAD mechanism and the ϵ-Greedy algorithm is shown in Figure 5.4. The ECAD mechanism stays consistent in minimizing regret as the number of end devices increases, highlighting its robustness and adaptability in dynamic environments. In contrast, the ϵ-Greedy algorithm exhibits an increase in total regret without any consistent pattern, indicating its inability to adapt to the changing environment effectively.

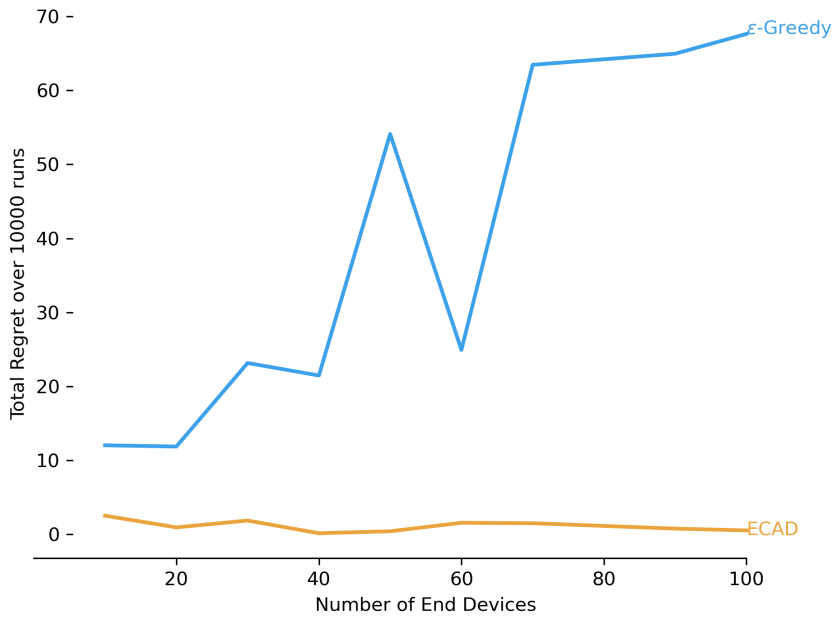

Figure 5.4: Total Regret versus Number of End Devices

5.1.4 Instantaneous and Cumulative Regret Comparison

The expected regret and the instantaneous regret are defined in Equations 4.3, and 4.4, respectively. Its was the difference between the power consumption of the optimal policy, and the power consumption of the policy that the relay is following. The optimal policy is the one that the relay only performs CAD process in the time slot that the desired end device is transmitting.

In this subsection, the instantaneous regret and cumulative regret comparison of the ECAD mechanism with the UCB, Greedy, ϵ-Greedy, and Thompson Sampling algorithms are presented. The comparison is made based on the average regret obtained from 1000 simulations.

5.1.4.1 Instantaneous Regret Comparison

We defined the reward as a periodic function in Equation (4.1), and the reward will be 1 if the end device is transmitting, and -1 if the fraud end device is transmitting. The instantaneous regret is calculated by taking the difference between the reward of

the optimal policy at a time instance, and the reward of the policy that the relay is following at a time instance. It was defined in Equation (4.4). The instantaneous regret comparison of the algorithms $ECAD$, Thompson Sampling, ϵ-Greedy, QGreedy is shown in Figure 5.5.

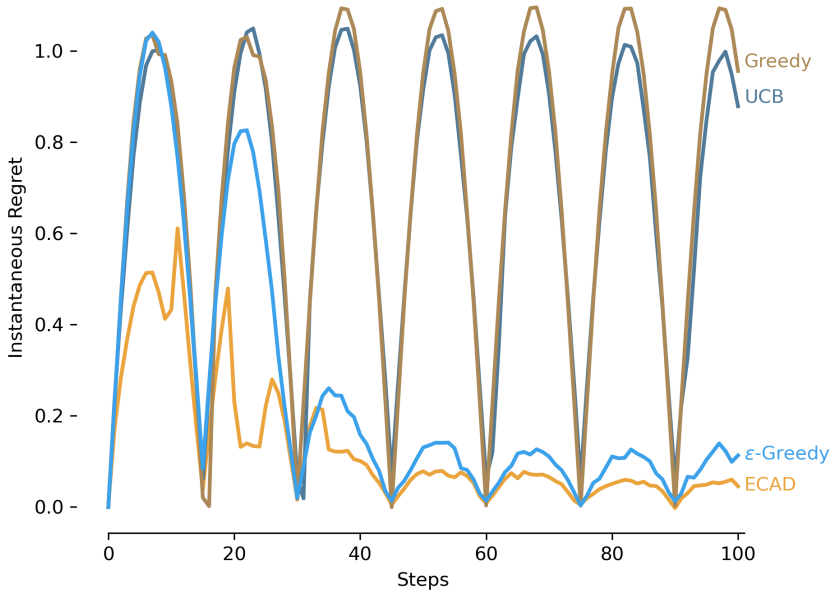

Figure 5.5: Instantaneous Regret Comparison

The ECAD mechanism exhibits a markedly superior performance when compared to the UCB, Greedy, ϵ-Greedy, and Thompson Sampling algorithms, specifically in terms of minimizing instantaneous regret. A significant aspect of the ECAD algorithm's performance is its ability to converge to the optimal arm at a much faster rate than its closest competitor, the ϵ-Greedy algorithm. This accelerated convergence underscores the ECAD mechanism's exceptional capability in swiftly and efficiently identifying and exploiting the most rewarding arms within the multi-armed bandit problem. The ECAD mechanism's proficiency in rapidly homing in on the optimal choices demonstrates its overall effectiveness in maximizing rewards while minimizing missed opportunities.

5.1.4.2 Cumulative Regret Comparison

The cumulative regret comparison is shown in Figure 5.6.

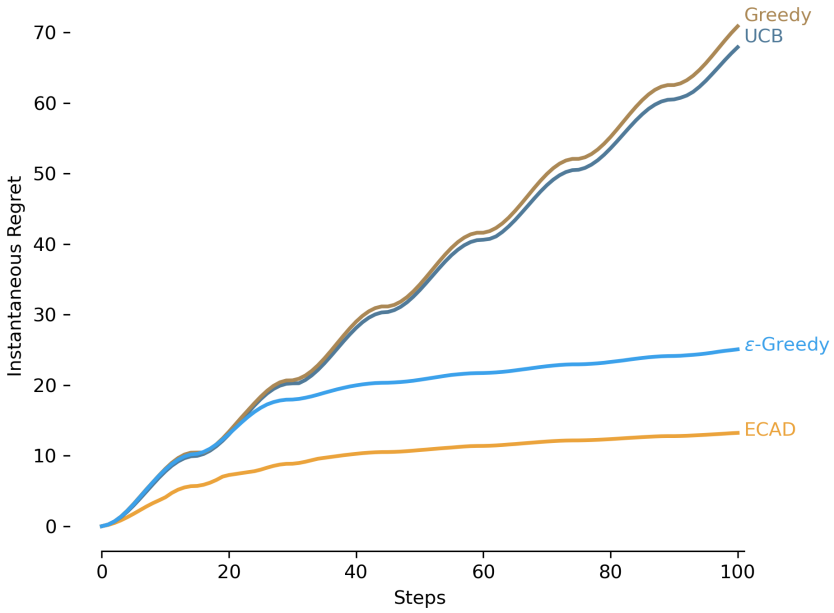

Figure 5.6: Cumulative Regret Comparison

The ECAD algorithm demonstrates the lowest instantaneous regret, indicating superior performance in minimizing regret. It consistently outperforms the other algorithms, converging more quickly to the optimal arm. In contrast, the Greedy algorithm exhibits the highest instantaneous regret, followed by UCB and ϵ-Greedy. The reason for the high regret values of the Greedy algorithm is its inability to explore and exploit the optimal arm effectively. Especially if the optimal arm is located at the end of the time slots, the Greedy algorithm will not be able to find it. The UCB algorithm, while more effective than the Greedy algorithm, still lags behind the ECAD algorithm in terms of minimizing regret. The ϵ-Greedy algorithm performs better than the UCB and Greedy algorithms but falls short of the ECAD algorithm in minimizing regret. This comparison highlights the efficiency and effectiveness of the ECAD algorithm in the multi-armed bandit problem.

5.1.5 Effect of the Number of Fraud End Devices

In this subsection, the impact of the number of fraud end devices on the total reward and total regret is analyzed. The fraud end device is defined as the end device that is not registered to the network, but the relay captures its uplink data and demodulates the message. The number of fraud end devices is varied from 1 to 30, while the number of empty scans is adjusted accordingly.

5.1.5.1 Total Reward Versus Number of Fraud End Devices

The total reward versus the number of fraud end devices is shown in Figure 5.7.

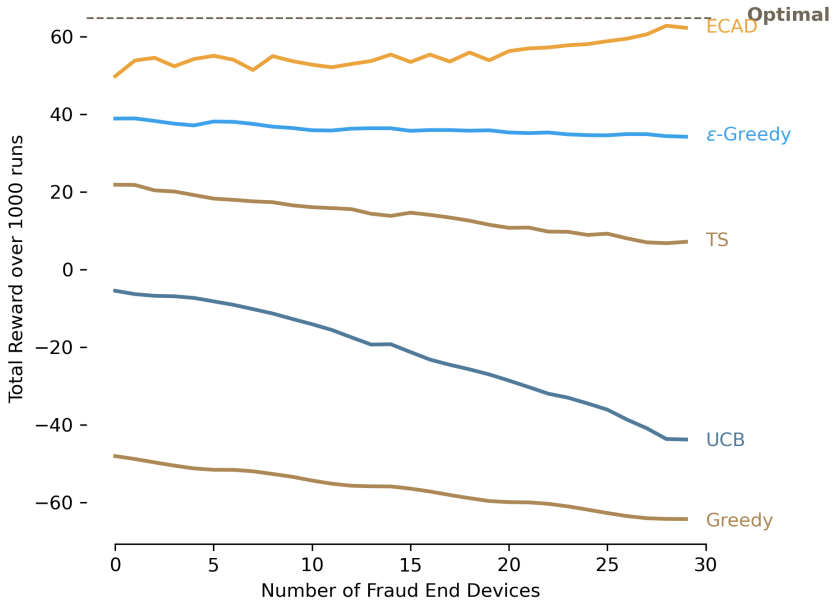

Figure 5.7: Total Reward Versus Number of Fraud End Devices

The algorithms instead of the ECAD mechanism cannot adapt their strategies to the changing environment, leading to a decrease in total reward as the number of fraud end devices increases. In contrast, the ECAD mechanism effectively identifies and avoids the fraud end devices, resulting in a consistent total reward across varying numbers of fraud end devices. This demonstrates the ECAD mechanism's robustness and adaptability in dynamic environments, enabling it to maintain high performance

levels regardless of the number of fraud end devices present.

5.1.5.2 Total Regret Versus Number of Fraud End Devices

The total regret versus the number of fraud end devices is shown in Figure 5.8.

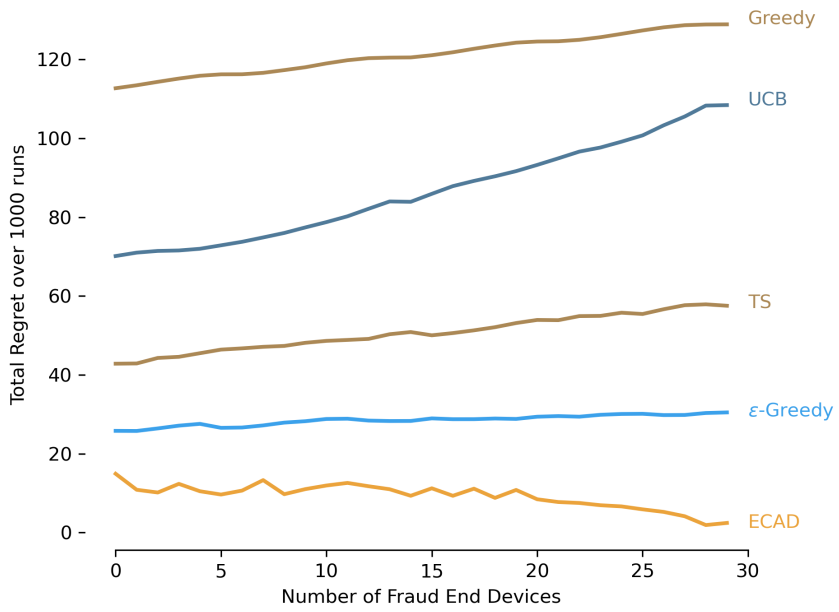

Figure 5.8: Total Regret Versus Number of Fraud End Devices

The ECAD mechanism demonstrates the lowest total regret across different numbers of fraudulent end devices, underscoring its superior performance in minimizing regret. In stark contrast, the UCB, Greedy, ϵ-Greedy, and Thompson Sampling algorithms exhibit a marked increase in total regret as the number of fraudulent end devices increases. This comparison vividly illustrates the ECAD mechanism's remarkable capability to adapt to dynamic environments and effectively reduce the negative impact of fraudulent end devices on overall network performance. The ECAD algorithm's adaptability and efficiency in handling varying levels of fraud emphasize its robustness and reliability in maintaining optimal performance under challenging conditions.

5.1.6 Comparison with CAD in LoRaWAN

The ECAD mechanism is evaluated in comparison with the CAD mechanism within the context of LoRaWAN™ networks, focusing on the number of packets successfully received over a given period. This comparison is conducted by varying both the number of end devices and the distance between these devices and the gateway. Additionally, the energy consumption of the end devices and relays during this period is taken into account.

This analysis underscores the effectiveness of the ECAD mechanism in distinguishing between optimal and fraudulent arms, thereby maximizing network performance while minimizing energy consumption. The ability to identify and prioritize the optimal arm, while filtering out noise and fraudulent arms, highlights the ECAD mechanism's superior operational efficiency and energy-saving potential in LoRaWAN™ networks.

To introduce this reward function to the real-life case, we can consider that the packets are sent at data rate SF9 and the bandwidth is 125 kHz. We can see that just by counting the number of extremum points in the reward function, we can find the number of transmission from end devices to relay at given time. We can count 14 extremum points in the reward function, which means that 14 packets are transmitted in a given time. It can be assumed that, intuitively, the 7 of them are from the correct end device, and the other 7 are from the fraud end device, the demonstration is given in Figure 5.9.

There are two end devices that are transmitting uplink packets periodically. It is considered that the one that gives a reward lower than zero (blue one), is not registered to the network. The red one is registered to the network, and it is the desired end device to demodulate and process its data.

The power consumption values are considered in three states, no activity, CAD activity only, CAD activity and demodulation. The values are taken from the datasheet of the LoRa radio module SX1262 from Semtech[3], and shown in Table 5.2.

In Figure 5.10, the power consumption and number of received packets are compared

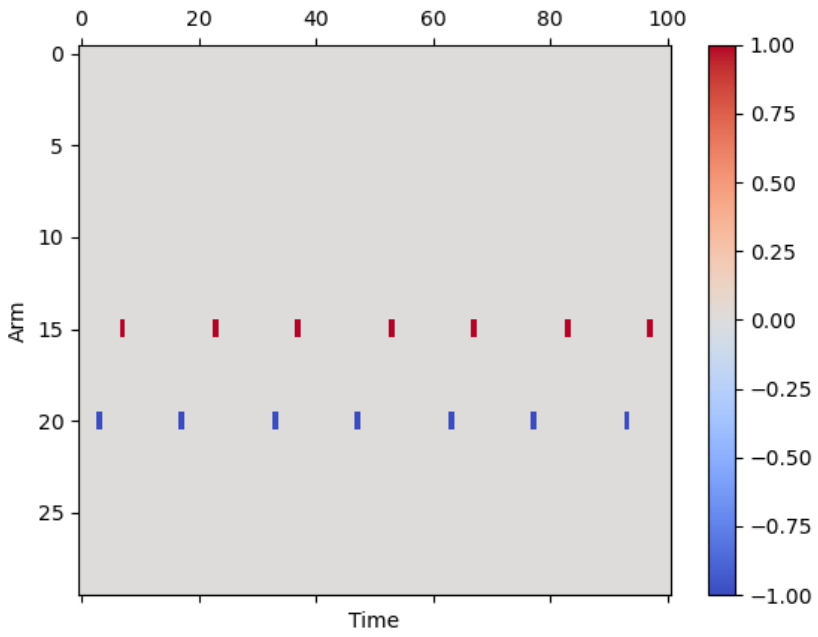

Figure 5.9: Maxima and Minima of the Reward Function

Table 5.2: Power Consumption Values for LoRa Radio Module SX1262[3] and STM32L476[4]

State	**Power Consumption SX1262**	**Power Consumption STM32L476**
No Activity	1.2 μ Ah	2μAh
CAD Activity	37.94 nAh	N/A
CAD Activity and Demodulation	4.6 mAh	40 mAh @ 80MHz

for the ECAD and CAD mechanisms. It is assumed that there are 300 time slots and an end device is transmitting data with a periodicity of 10 time slots. The aim of this comparison is to find the number of packets that are received by the relay, and the power consumption of the relay during this period. There is only one registered end device that is transmitting data, other time slots includes only noise, which means that if the relay listen another time slot, it will get noise at the CAD activity.

In Figure 5.10, the ECAD mechanism in terms of received packets, the ECAD mech-

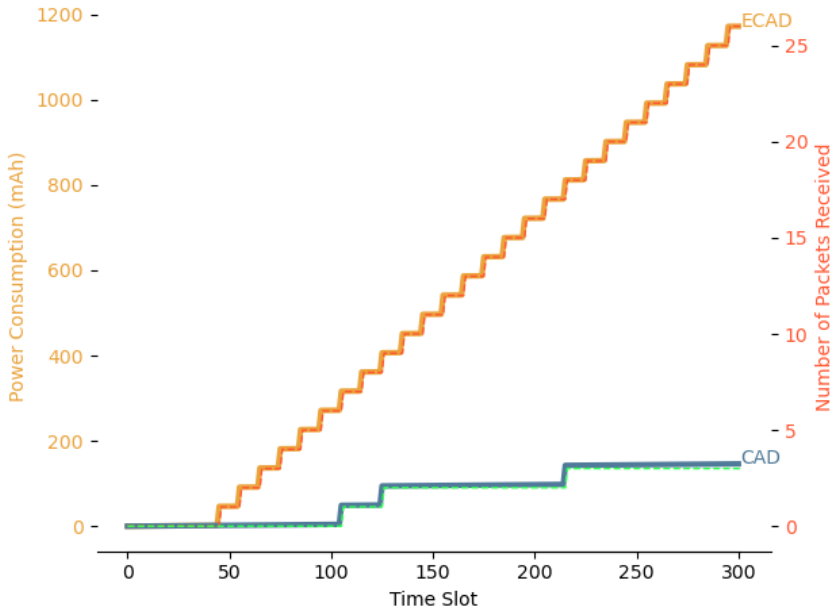

Figure 5.10: Power Consumption of ECAD and CAD Mechanisms

anism learns the periodicity of the end device until the 50th time slot. After the 50th time slot, the ECAD mechanism starts to demodulate the signal, and the number of received packets increases. The CAD mechanism, on the other hand, listens to all time slots, and the number of received packets is the same as the ECAD mechanism until the 50th time slot. After the 50th time slot, the CAD mechanism starts to demodulate the signal, and the number of received packets increases. However, CAD mechanism only receives 4 packets, while the ECAD mechanism receives 29 packets.

The reason that ECAD mechanism consumes more power than the CAD mechanism is that it has found the periodicity of the end device after 50th slot and demodulates each packet that is transmitted by the end device.

This efficiency arises because the ECAD mechanism exclusively receives packets from the correct end devices, whereas the CAD mechanism processes all incoming packets and attempts to demodulate them indiscriminately. By filtering out fraudulent end devices and disregarding empty scans, the ECAD mechanism achieves a more efficient and energy-saving operation. This heightened energy efficiency represents a critical advantage of the ECAD mechanism, rendering it a more sustainable and cost-

effective solution for LoRaWAN™ networks. The reduced power consumption not only lowers operational costs but also contributes to the environmental sustainability of network operations, highlighting the ECAD mechanism's superior performance in both energy management and overall system efficency.

We said that, the ECAD algorithm enables the relay to support more than 10 end devices, intuitively, this is because the relay does not need any additional messaging sequences between any parties and it just listens for an uplink packet coming from and end devices and encapsulates and sends it. The relay gets the device list from the gateway when it is registered to the network.

5.2 LoRa End Device PCB Design

In this section, the PCB design of a LoRa end device is presented. The LoRa end device PCB design consists of two main components: the breakout board and the base board. The breakout board is designed to interface with the LoRa module, while the base board provides additional functionalities such as power management, sensor interfacing, and communication interfaces.

5.2.1 Bill of Materials

The reduced set of components used in the LoRa end device PCB design is listed in Table 5.3.

The LoRa end device PCB design incorporates a minimal set of components to ensure cost-effectiveness and simplicity. The key components include the LoRa module, microcontroller, voltage regulator, LED, push button, RTC module, battery, and charge module. These components are carefully selected to provide essential functionalities while maintaining a compact and efficient design.

Table 5.3: Bill of Materials for LoRa End Device PCB Design

Component	Quantity	Description
LoRa Module	1	RAK3172
Microcontroller	1	ESP32
Voltage Regulator	1	AMS1117
Voltage Regulator	1	RT9080
LED	1	Red
Button	1	Push Button
RTC Module	1	DS1307
Battery	1	LiPo 3.7V
Chage Module	1	TP4056

5.2.2 Schematic Explanation

5.2.2.1 Breakout Board

The schematic design of the breakout board is shown in Figure A.1.

The breakout board schematic design features the LoRa module, an antenna, a voltage regulator for standalone operation, and communication interfaces for the base board or master devices. The breakout board is designed to provide a simple and reliable interface for the LoRa module, enabling seamless integration with the base board.

5.2.2.2 Base Board

The schematic design of the base board is shown in Figure A.2.

The base board is designed to control the modem, which is in the breakout board, and to provide additional functionalities such as power management, sensor interfacing, and communication interfaces.

5.2.3 PCB Layout Explanation

5.2.3.1 Breakout Board

The PCB layout design of the breakout board is shown in Figure A.3.

The breakout board PCB layout design features a compact and efficient layout to accommodate the LoRa module, antenna, voltage regulator, and communication interfaces. The breakout board is designed to be easily integrated with the base board, providing a seamless interface for the LoRa module. The PCB board is 2 layered with dimensions lower than 100x100 mm. The breakout has a SMA 868 MHz antenna connector, 2 pin header Universal Asynchronous Receiver-Transmitter (UART) ports (3.3V logic), a surface mounted device (SWD) port, a programming switch, 2 power inputs (3.3V and 5V), and 2 buttons for reset and boot.

5.2.3.2 Base Board

The base board is designed to control multiple breakout boards, if needed. The orientation of the breakout board on the base board can be figured out just by considering the power pins, they should be matched both on breakout and relay boards. The front and back side of the base board PCB layout design is shown in Figure A.4 and Figure A.5, respectively.

First breakout board fits directly to the board, and the second one is connected by UART from $2NDRAKUARTport$ on the base board.

The base board includes:

- 1 CH340 USB to UART converter
- 1 ESP32 WROOM Microcontroller
- 1 TP4056 Charge Module
- 2 RT9080 Voltage Regulator (3.7V to 3.3V), second one is optional in case of the second breakout board

- 1 DS1307 RTC Module

From Relay board, serial peripheral interface (SPI) and UART ports of the fitted breakout board can be accessible

5.3 Patent Application

In this section, the patent application for the proposed relay device is presented. The patent application includes a detailed description of the relay device, its components, and its operation. The patent application is designed to protect the intellectual property rights of the proposed relay device and ensure its commercial viability. The application is made to the Turkish Patent and Trademark Office (TPTO) and the World Intellectual Property Organization (WIPO). There is a PCT patent application incentive program, provided by TUBITAK, the application is made within this program. The PCT application is made, the application number is $PCT/TR2023/051024$. It is in the search report phase, and the results are expected to be announced in the upcoming months.

CHAPTER 6

CONCLUSIONS

We investigated a conceptual design of a tunneling relay device which is the enhanced version of the existing LoRaWAN™ relay device. The primary goal of the relay device is to provide LoRaWAN™ connectivity to end devices suffering from poor channel conditions to reach the gateway. There is an existing relay standard defined by the LoRa Alliance™ that forwards uplink packets from end devices to the gateway. However, this relay device has some limitations, such as supporting up to 10 end devices and requiring a long preamble to communicate with an end device. This can lead to increased power consumption and reduced network capacity. The enhancement of the relay device is proposed in this thesis work to address these limitations. The proposed relay device supports more end devices, reduces power consumption, and improves network capacity.

Next, we investigated the multi armed bandit problem in the context of LoRaWAN™ relay devices. The ECAD mechanism was proposed to detect uplink frames from end devices with a low probability of false alarm and a high probability of detection. Then, the ECAD mechanism is compared with the existing CAD mechanism, and the existing multi armed bandit algorithms. The simulation results show that the ECAD mechanism outperforms the CAD mechanism and the existing multi armed bandit algorithms in terms of the probability of detection and the probability of false alarm. The ECAD mechanism is also evaluated in terms of the average reward, average regret, the instantaneous regret, and the results show that the ECAD mechanism achieves a higher average reward compared to the existing multi armed bandit algorithms. The ECAD mechanism is also tested under increasing number of fraud devices in the network, and the results show that the ECAD mechanism is robust

against fraud devices.

In future works, the proposed relay device can be implemented in a real-world scenario to evaluate the performance of the relay device. The proposed ECAD algorithm can be further improved by considering the cases different than periodic transmissions. The proposed relay device can be tested in a real-world scenario to evaluate the performance of the relay device. The power consumption of the relay device can be measured in a real-world scenario to compare it with the existing relay device. If the benefits of the proposed relay device are confirmed in a real-world scenario, the proposed relay device can be standardized by the LoRa Alliance™.

REFERENCES

[1] O. Gencay, A. Balci, E. T. Ceran, and E. Uysal, "Ensuring a semantically effective iot network through blockchain with delayed feedback," in *2024 International Conference on Computing, Networking and Communications (ICNC)*, pp. 88–92, 2024.

[2] Semtech, "An1200.22, lora modulation basics." https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001OJa/2BF2MTeiqIwkmxkcjjDZzalPUGlJ76lLdqiv.30prH8, 2015.

[3] Semtech, "Sx1262 datasheet." https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1262.

[4] STMicroelectronics, "Stm32l4 series." https://www.st.com/en/microcontrollers-microprocessors/stm32l4-series.html.

[5] D.-H. Kim, E.-K. Lee, and J. Kim, "Experiencing lora network establishment on a smart energy campus testbed," *Sustainability*, vol. 11, p. 1917, 03 2019.

[6] L. Alliance, "Rp002-1.0.4 regional parameters." https://resources.lora-alliance.org/technical-specifications/rp002-1-0-4-regional-parameters, 2022.

[7] L. Alliance, "Ts001-1.0.4 l2 specification." https://resources.lora-alliance.org/document/ts001-1-0-4-lorawan-l2-1-0-4-specification, 2020.

[8] L. Alliance, "Lorawan relay." https://resources.lora-alliance.org/technical-specifications/ts011-1-0-0-relay.

[9] Semtech, "Lora technology." https://www.semtech.com/lora.

[10] L. Alliance, "Lora alliance." https://lora-alliance.org/.

[11] L. Alliance, "Lorawan." https://lora-alliance.org/lorawan.

[12] I. Analytics, "Number of connected iot devices growing 16% to 16.7 billion globally," 2023.

[13] E. Calvanese Strinati and S. Barbarossa, "6g networks: Beyond shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107930, 2021.

[14] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, B. Soret, *et al.*, "Semantic communications in networked systems: A data significance perspective," *IEEE Network*, vol. 36, no. 4, pp. 233–240, 2022.

[15] Y. Guo, S. Zhang, and D. Xiao, "Overview of wi-fi technology," *The 2nd International Conference on Computer Application and System Modeling*, 08 2012.

[16] J. Haartsen, "Bluetooth - the universal radio interface for ad hoc, wireless connectivity," *Ericsson review*, vol. 75, pp. 110–117, 01 1998.

[17] N. Somani and Y. Patel, "Zigbee: A low power wireless technology for industrial applications," *International Journal of Control Theory and Computer Modeling*, vol. 2, 05 2012.

[18] B. Prabhakar, E. Uysal Biyikoglu, and A. El Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 1, pp. 386–394 vol.1, 2001.

[19] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal, "Energy-efficient packet transmission over a wireless link," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 487–499, 2002.

[20] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[21] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[22] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, pp. 285–294, 1933.

[23] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions, Volume 2*. New York: Wiley, 1995.

[24] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[25] Y. Wang, "Review on greedy algorithm," *Theoretical and Natural Science*, vol. 14, pp. 233–239, 11 2023.

[26] A. dos Santos Mignon and R. L. de Azevedo da Rocha, "An adaptive implementation of ϵ-greedy in reinforcement learning," *Procedia Computer Science*, vol. 109, pp. 1146–1151, 2017.

[27] M. R. WINKLER, "Chirp signals for communications," in *IRE Wescon Conv Rec (IRE Wescon Convention Record)*, vol. 6, pp. 14.2/1–4, 1962.

[28] W. Hirt and S. Pasupathy, "Continuous phase chirp (cpc) signals for binary data communication-part i: Coherent detection," *IEEE Transactions on Communications*, vol. 29, no. 6, pp. 836–847, 1981.

[29] W. Hirt and S. Pasupathy, "Continuous phase chirp (cpc) signals for binary data communication - part ii: Noncoherent detection," *IEEE Transactions on Communications*, vol. 29, no. 6, pp. 848–858, 1981.

[30] M. Kowatsch and J. Lafferl, "A spread-spectrum concept combining chirp modulation and pseudonoise coding," *IEEE Transactions on Communications*, vol. 31, no. 10, pp. 1133–1142, 1983.

[31] R. Pickholtz, L. Milstein, and D. Schilling, "Spread spectrum for mobile communications," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 2, pp. 313–322, 1991.

[32] A. Springer, W. Gugler, M. Huemer, R. Koller, and R. Weigel, "A wireless spread-spectrum communication system using saw chirped delay lines," *IEEE*

Transactions on Microwave Theory and Techniques, vol. 49, no. 4, pp. 754–760, 2001.

[33] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[34] A. Lavric, A. I. Petrariu, and V. Popa, "Long range sigfox communication protocol scalability analysis under large-scale, high-density conditions," *IEEE Access*, vol. 7, pp. 35816–35825, 2019.

[35] Ingenu, "Rpma technology." https://www.ingenu.com/technology/rpma/.

[36] Telensa, "Telensa: The smartest choice for street lighting." https://www.telensa.com/.

[37] Qowisio, "Qowisio: Next generation iot solutions maker." https://www.qowisio.com/?lang=en.

[38] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," in *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, pp. 1–5, 2016.

[39] B. Vejlgaard, M. Lauridsen, H. Nguyen, I. Z. Kovacs, P. Mogensen, and M. Sorensen, "Coverage and capacity analysis of sigfox, lora, gprs, and nb-iot," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pp. 1–5, 2017.

[40] J. Petäjäjärvi, K. Mikhaylov, M. Hämäläinen, and J. Iinatti, "Evaluation of lora lpwan technology for remote health and wellbeing monitoring," in *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, pp. 1–5, 2016.

[41] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[42] H. Mroue, A. Nasser, B. Parrein, S. Hamrioui, E. Mona-Cruz, and G. Rouyer, "Analytical and simulation study for lora modulation," in *2018 25th International Conference on Telecommunications (ICT)*, pp. 655–659, 2018.

[43] M. R. Seye, B. Ngom, B. Gueye, and M. Diallo, "A study of lora coverage: Range evaluation and channel attenuation model," in *2018 1st International Conference on Smart Cities and Communities (SCCIC)*, pp. 1–4, 2018.

[44] L. Alliance, "Ts002-1.1.0 backend interfaces." https://resources.lora-alliance.org/technical-specifications/ts002-1-1-0-lorawan-backend-interfaces, 2020.

[45] Semtech, "Sx1302 datasheet." https://www.semtech.com/products/wireless-rf/lora-core/sx1302.

[46] H. T. Friis, "A note on a simple transmission formula," 05 1946.

[47] E. T. S. I. (ETSI), "Short range devices (srd) operating in the frequency range 25 mhz to 1 000 2101 mhz; part 2: Harmonized standard for access to radio spectrum for non-specific radio 2102 equipment, v.3.2.1," 2019.

[48] H. Taub and D. L. Schilling, "Principles of communication systems," 1986.

[49] S. Corporation, "Lora modulation basics," 2015.

[50] G. King, M. Roland-Mieszkowski, T. Jason, *et al.*, "Noise levels associated with urban land use," *Journal of Urban Health*, vol. 89, no. 6, pp. 1017–1030, 2012.

[51] S. Corporation, "Application note: Sx126x cad performance evaluation." https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/3n000000qSr3/s723KzQi7zXFhBPKqesoF11.MtwYu6B8pdEQcmInlkE, 2019.

[52] H. Cai, Z. Cen, L. Leng, and R. Song, "Periodic-gp: Learning periodic world with gaussian process bandits," *arXiv preprint arXiv:2105.14422*, 2021.

[53] N. Srinivas, A. Krause, and S. M. Kakadet, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2010.

[54] N. Chen, C. Wang, and L. Wang, “Learning and optimization with seasonal patterns,” *Operations Research*, 2023.

APPENDIX A

PRINTED CIRCUIT BOARD DESIGNS

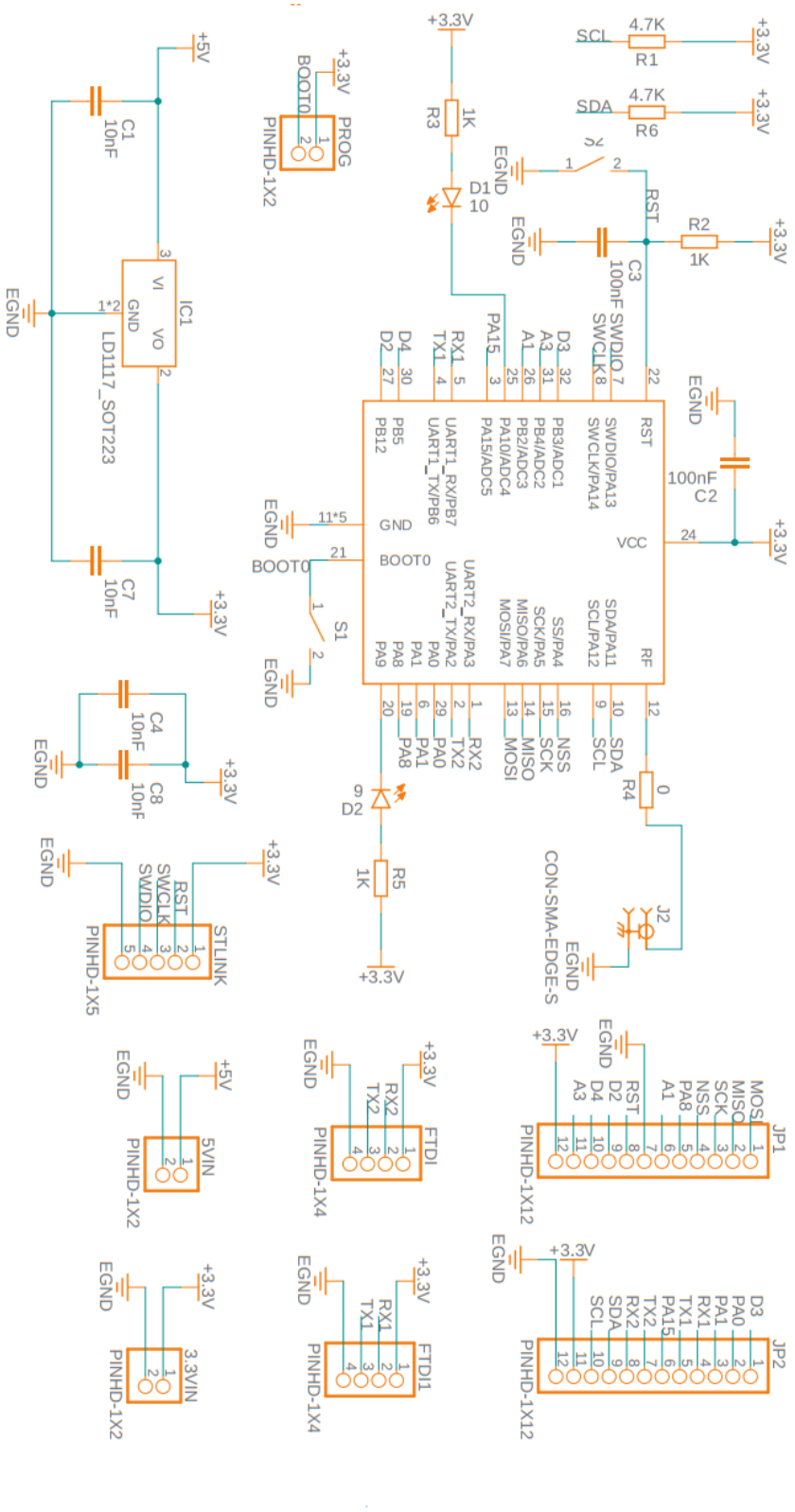

Figure A.1: Breakout Board Schematic Design

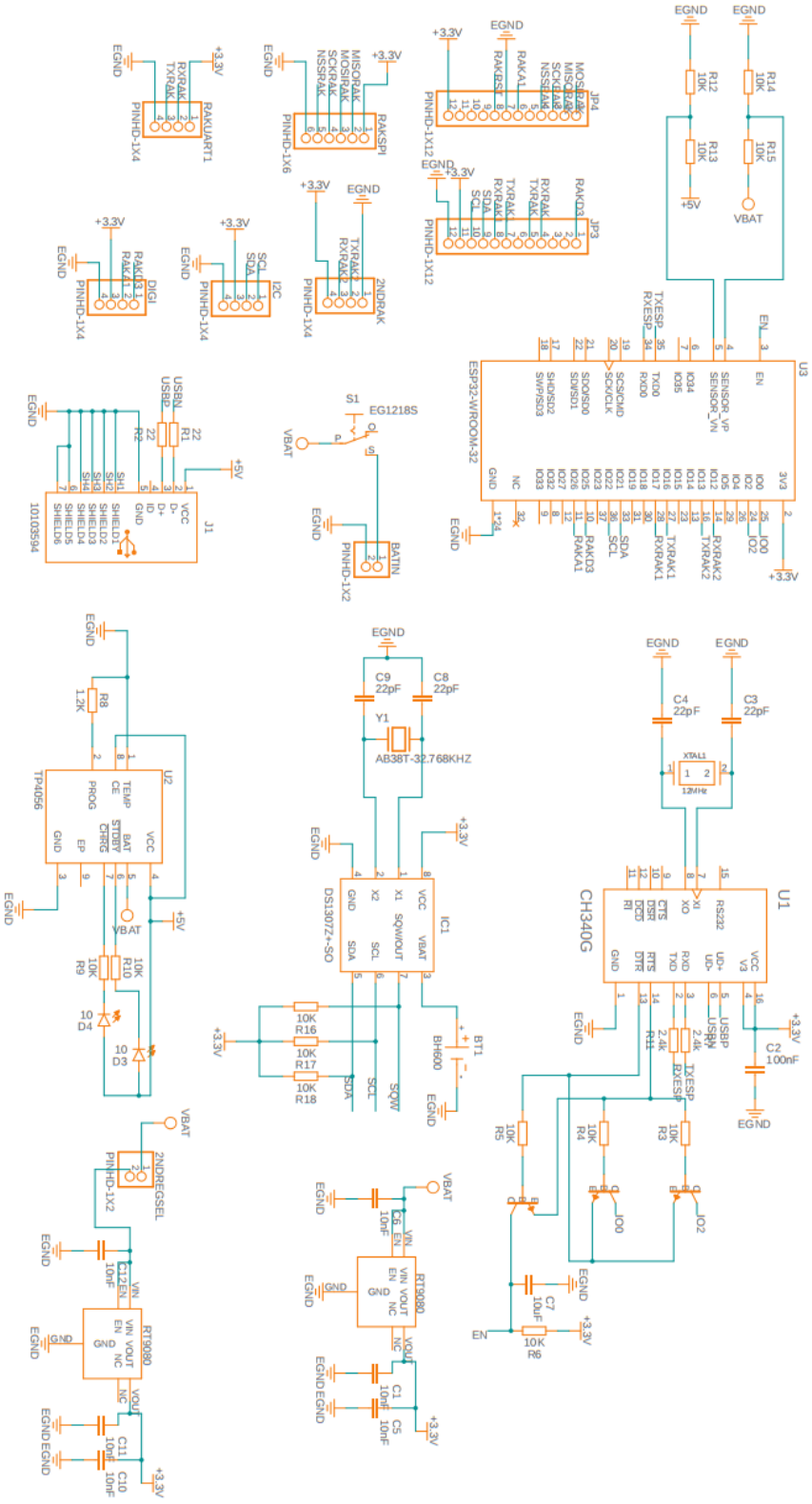

Figure A.2: Base Board Schematic Design

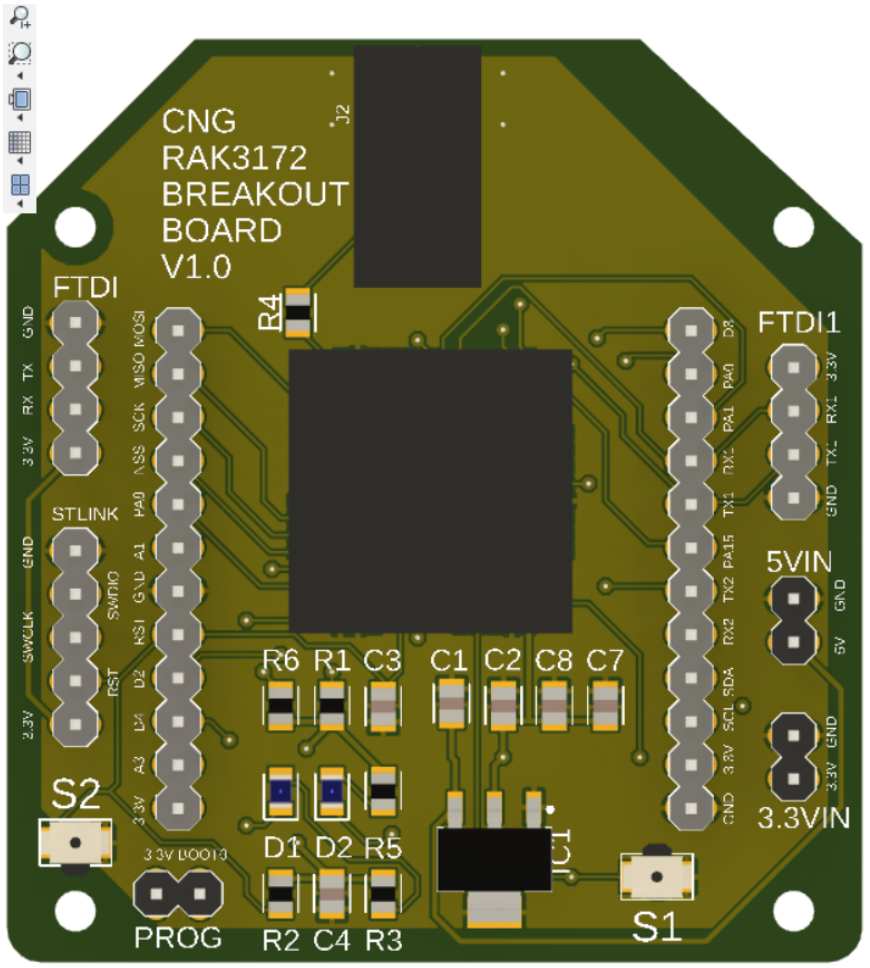

Figure A.3: Breakout Board PCB Layout Design

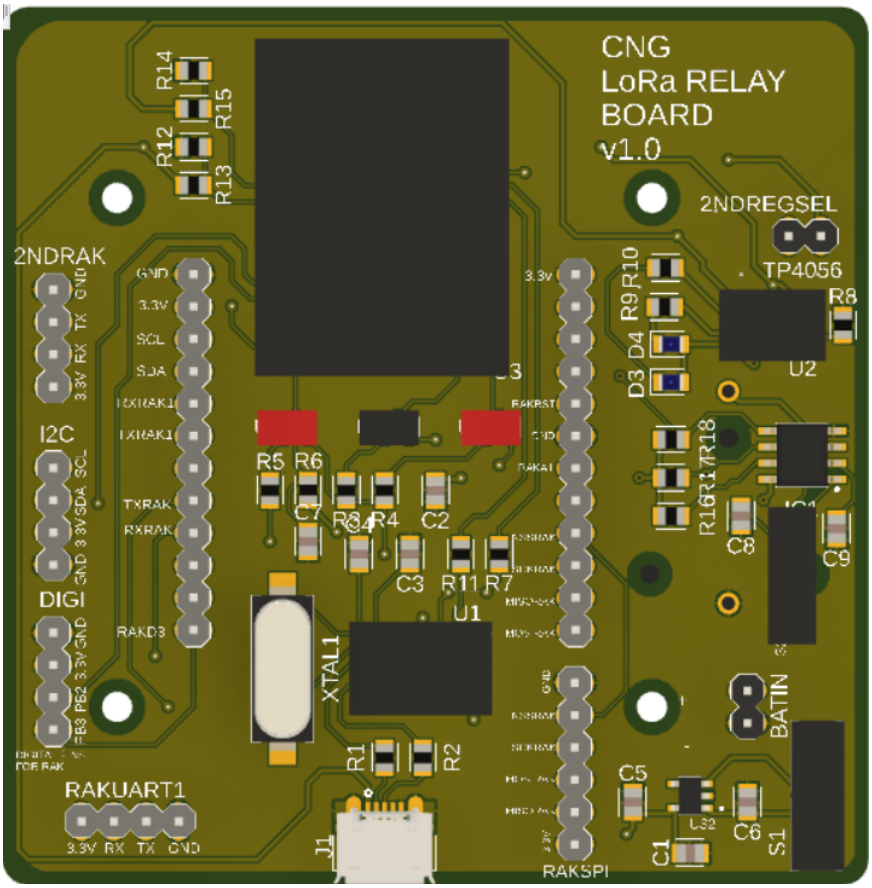

Figure A.4: Base Board PCB Layout Design (Front Side)

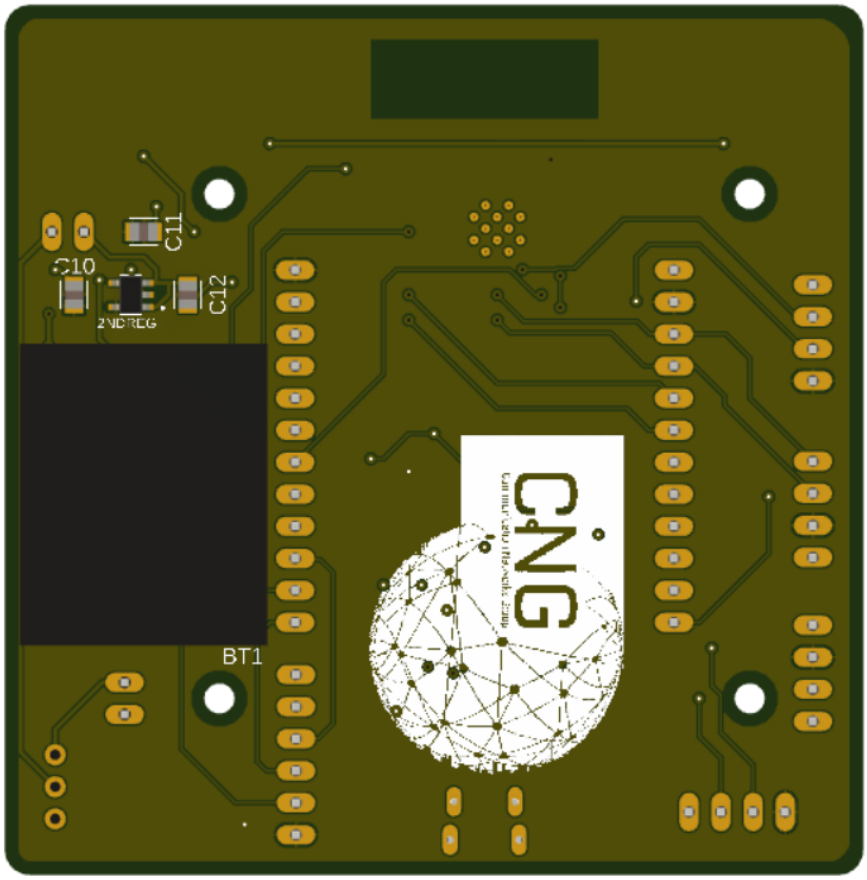

Figure A.5: Base Board PCB Layout Design (Back Side)