Title
A Modular Framework for Reliable Throughput and Time-Synchronization in Low Power WSNs

Author
Shah, Sureel

Publication Date
2024

UNIVERSITY OF CALIFORNIA SAN DIEGO

A Modular Framework for Reliable Throughput and Time-Synchronization in Low Power WSNs

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Sureel Bhadreshbhai Shah

Committee in charge:

Professor Dinesh Bharadia, Chair
Professor Nikolay Atanasov
Professor Patrick Pannuto

2024

The thesis of Sureel Bhadreshbhai Shah is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I am profoundly grateful to the many individuals who have supported and guided me throughout this remarkable journey.

Chapter 4, in part, is a reprint of the material as it appears in ”XRLoc: Accurate UWB Localization to Realize XR Deployments” by Aditya Arun, Shunsuke Saruwatari, Sureel Shah, which was published at Conference on Embedded Network Sensor Systems (SENSYS), 2023. This thesis author was the co-primary investigator and the author of this paper.

VITA

2019	Bachelor of Science, State University of New York at Buffalo
2024	Master of Science, University of California San Diego

ABSTRACT OF THE THESIS

A Modular Framework for Reliable Throughput and Time-Synchronization in Low Power WSNs

by

Sureel Bhadreshbhai Shah

Master of Science in Electrical Engineering (Intelligent Systems, Robotics, and Control)

University of California San Diego, 2024

Professor Dinesh Bharadia, Chair

The proliferation of Wireless Sensor Networks (WSNs) has necessitated advances in energy-efficient and highly synchronized communication protocols to support the diverse range of Internet of Things (IoT) applications. This thesis presents a modular framework designed to address the critical challenges of reliable throughput, time synchronization, and energy efficiency in low-power WSNs. The framework leverages LoRa-based hardware to implement a novel Medium Access Control (MAC) protocol that facilitates precise time synchronization and robust data scheduling. Key contributions include the development of a one-shot synchronization mechanism capable of aligning thousands of nodes with sub-microsecond accuracy, a TDMA-based scheduling mechanism, and the implementation of energy-efficient techniques such as Low Power Listening (LPL) and duty cycling. This work not only enhances the performance and reliability of WSNs, but also provides a flexible platform adaptable to various IoT scenarios, including environmental monitoring, power grid monitoring, and real-time location systems.

Chapter 1

Introduction

1.1 Background

The recent advancements in technology related to micro-electro-mechanical systems have accelerated the progress of small, cost-effective, energy-efficient, and versatile sensing devices, which have the capability to carry out various functions like sensing, processing data, and communicating [5] [6]. A wireless sensor network (WSN) is a decentralized network typically composed of numerous sensor nodes, that are distributed in space and engage in interactions to establish a network. [8] These networks exhibit resilience, flexibility towards modifications, and a high degree of programmability. Consequently, they constitute a fundamental component of an intelligent urban environment for executing diverse functionalities. [7]. The main motivation for deploying a sensor network, comprising numerous battery-powered nodes, is the need for remote and unobtrusive monitoring.

Wireless sensor networks have a multitude of applications across different fields such as healthcare [36], industry [38], military [35], environmental monitoring [37], science and structural health monitoring to name a few. More specifically in the medical field, WSNs enable healthcare professionals to remotely track patient's physiological signals to detect specific symptoms, with health wearables widely adopted at a consumer level. In military operations, WSNs help in identifying nuclear, biological, and chemical hazards, detecting hazardous materials, alerting about enemy intrusions by recognizing enemy units, and monitoring friendly troops, equipment, and armaments. In addition, WSNs are advantageous for tracking wildfires, analyzing ecological and biological habitats, sensing natural disasters such as floods and earthquakes, and monitoring air quality. In everyday civilian life, WSNs facilitate finding available parking spots in public areas, monitoring employee identification badges in workplaces, bolstering security in public venues like banks and stadiums, and managing traffic flow on highways during certain times. Additionally, they can enhance smart home systems by automating energy usage, improving health monitoring for the elderly, and providing real-time data for urban planning and development.

In wireless sensor networks (WSNs), the autonomous operation of individual sensors often leads to a critical challenge: the lack of synchronization among their internal clocks. This can significantly complicate the integration and interpretation of data collected from various nodes distributed throughout the network. For example, when a moving vehicle is detected at different locations, the precise timing of these detections becomes vital for accurately assessing the vehicle's trajectory. Without a synchronized time reference, comparing the detection timestamps can yield misleading conclusions about the vehicle's direction of travel.

Additionally, scenarios involving cooperative sensing, such as the establishment of beamforming arrays or the implementation of Time Division Multiple Access (TDMA) radio scheduling, necessitate that nodes operate on a common time frame. In these cases, synchronization becomes pivotal to ensure that multiple nodes can effectively coordinate their actions and share data in a coherent manner.

These various scenarios emphasize the critical need for a synchronized time concept within wireless sensor networks. Furthermore, achieving synchronization can be challenging due to factors such as network latency, clock drift, and environmental interference, which can all impact the reliability of time-sensitive data transmission. Consequently, there has been a growing interest in the development of energy-efficient clock synchronization protocols and Medium Access Control (MAC) protocols aimed at fostering a shared sense of time across the network. Such advancements not only enhance the accuracy of data interpretation and localization but also improve the overall efficiency and reliability of the wireless sensor network's operations.

1.2 Motivation

In wireless sensor networks (WSNs), the autonomous operation of individual sensors presents a significant challenge, the lack of synchronization among their internal clocks. This desynchronization can severely complicate the integration and interpretation of data collected from various nodes dispersed throughout the network. When sensors operate on unsynchronized clocks, the timestamps associated with the data they generate may not align accurately, which poses a considerable obstacle for analyzing events or phenomena that rely on temporal relationships. For instance, in applications such as environmental monitoring or disaster response, precise timing is crucial for correlating sensor readings and deriving meaningful insights. Consequently, inaccuracies in clock synchronization can lead to data integrity issues, where the reliability of the information is undermined, making it difficult to establish cause-and-effect relationships or to track the evolution of events over time. The ramifications of this fundamental issue extend beyond mere data interpretation; they also manifest in challenges related to scheduling within the network. Accurate synchronization is vital for implementing effective Time Division Multiple Access (TDMA)-based radio scheduling. Without synchronized clocks, nodes are compelled to engage in energy-intensive channel sensing to coordinate their transmissions. This reliance on unscheduled and uncoordinated communication methods not only leads to inefficient utilization of the wireless medium but also results in increased packet collisions and higher retransmission rates. Such inefficiencies can significantly deplete battery life, which is particularly critical in WSNs, where nodes are often deployed in remote locations with limited access to power sources. Moreover, the issues of time synchronization and scheduling are intricately linked to the scalability of the network. As WSNs expand, the ability to maintain effective time synchronization and scheduling becomes increasingly important. Synchronized scheduling minimizes collisions and optimizes the use of the communication channel, which is essential for accommodating the growing number of nodes. This structured approach enhances overall network integrity and performance, ensuring that the integration of new nodes does not compromise the efficiency and reliability of the system. Therefore, robust time synchronization and precise scheduling are not merely beneficial but essential for managing larger numbers of nodes in expanding WSNs. By facilitating efficient communication and minimizing energy consumption, these mechanisms play a pivotal role in enhancing the scalability and sustainability of wireless sensor networks.

1.3 Research Content

The primary objective of this thesis is to develop a modular framework that enhances the performance and efficiency of wireless sensor networks (WSNs) by using a low-power LoRa-based hardware platform. The motivation for choosing LoRa are delved into in the next section. This framework is designed to address key challenges in WSNs, such as time synchronization, reliable throughput, and energy efficiency. The research content is structured around the following main components:

- Time-Synchronization Techniques: The thesis designs and implements a novel method for achieving high-precision time-synchronization in WSNs. This includes a one-shot synchronization mechanism that can potentially synchronize thousands of nodes with an accuracy $\pm 20\mu s$. This is critical for ensuring the accurate coordination of sensor nodes and the reliable aggregation of data across the network.
- Modular MAC Protocol: A central part of the research is the design and implementation of a versatile MAC protocol that can be adapted to various WSN use cases. This protocol

incorporates features such as slotting and scheduling of motes to provide reliable throughput, side channeling for secondary communication paths, and energy-efficient communication strategies. The modularity of the protocol allows it to be used standalone or integrated with other systems.

- Energy Efficiency Improvements: Given the resource-constrained nature of WSN nodes, improving energy efficiency is paramount. The research explores innovative scheduling and slotting methods that reduce energy consumption without compromising network performance. These methods are tailored to the low power capabilities of the LoRa-based hardware platform.

- System Integration and Use Cases: The framework is designed to be flexible and adaptable, supporting integration with other WSN applications. The thesis includes a detailed case study of using the framework in an Ultra-Wideband Real-Time Locating System (UWB RTLS), demonstrating its practical application and benefits.

- Experimental Validation: To validate the effectiveness of the proposed framework, extensive testing and evaluation are conducted. This involves setting up a controlled experimental environment to measure the performance of the time-synchronization techniques, the MAC protocol, and the overall system efficiency.

1.4 Structure of the Thesis

The thesis is organized into five main chapters, each focusing on different aspects of the research. Below is an outline of the thesis structure:

- Introduction: This chapter provides the research background, a review of research status, an overview of the research content, and the structure of the thesis.

- Foundational Knowledge: This chapter lays the groundwork by discussing the fundamental concepts related to WSNs, time-synchronization techniques, MAC protocols, and LoRa technology. It provides the necessary context for understanding the subsequent chapters.
- Design and Implementation of the Modular Framework: This chapter delves into the detailed design and implementation of the proposed framework. It covers the system architecture, the development of time-synchronization methods, slotting and scheduling techniques, power efficiency improvements, and the features of the versatile MAC protocol.
- Experimental Methods and Results: This chapter describes the experimental setup, the process of conducting experiments, and the analysis of the results. It includes a comparison of the proposed framework with traditional MAC protocols to demonstrate its advantages.
- Conclusion and Outlook: The final chapter summarizes the key findings of the research, highlights the novel contributions made, and discusses potential directions for future work.

Chapter 2

Background

2.1 LPWANs

Low Power Wide Area Networks (LPWANs) have emerged as a pivotal component in the establishment and operation of large-scale Wireless Sensor Networks (WSNs). The significance of LPWANs stems from their ability to address one of the primary challenges in sensor network design: energy consumption. As sensor networks are often deployed in remote or hard-to-reach locations, the efficiency of energy usage becomes critical, directly impacting the longevity and effectiveness of the network. LPWANs facilitate long-range communication capabilities while simultaneously ensuring minimal power consumption and reduced deployment costs. This is in-part attained through creating a star network topology that enables devices to communicate with the gateway without any relaying of messages [1]. This unique combination makes them exceptionally suited for a diverse array of applications, including but not limited to smart city infrastructure [40], precision agriculture [39], environmental monitoring [41], and various Internet of Things (IoT) implementations.

In contrast, short-range wireless technologies such as Wi-Fi and Bluetooth, while effective for certain applications, frequently encounter limitations in scenarios that demand extensive

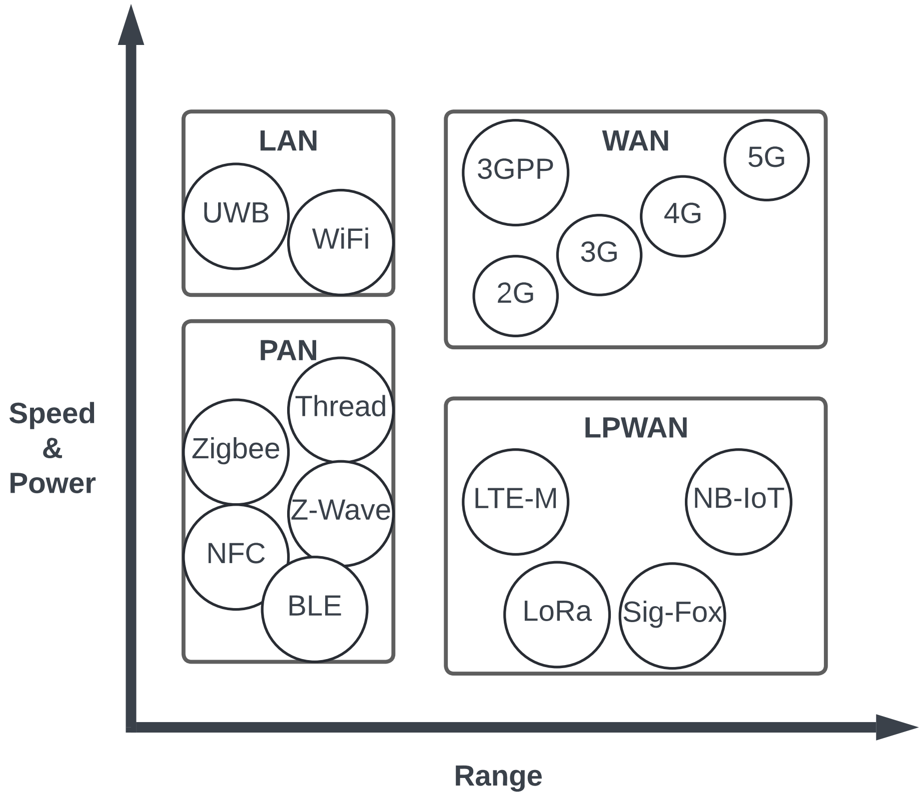

Figure 2.1: Wireless Communication Technologies Comparison [1]

coverage and prolonged battery life. These technologies are typically designed for high-speed data transfer over shorter distances, which can be a disadvantage in environments where devices are spread over large geographic areas. To illustrate the differences in performance across various wireless technologies, a comparative analysis focusing on speed, power consumption, and operational range is provided in Fig 2.1. This figure serves to highlight the advantages of LPWANs in terms of their ability to sustain long-distance communication with minimal energy expenditure, thereby reinforcing their role as a foundational technology in the evolution of modern sensor networks.

One of the primary challenges associated with LTE-M and NB-IoT is the reliance on existing cellular infrastructure, which may not be available in remote or underserved areas. This limitation can hinder the deployment of these technologies in locations where connectivity is essential but infrastructure is lacking. Additionally, the costs associated with leveraging cellular networks can be prohibitive, especially for large-scale sensor network implementations where budget constraints are a significant consideration. Thus, they are not a viable solution for this framework implementation.

LoRa and Sigfox offer alternative approaches that utilize unlicensed spectrum, enabling broader coverage and lower deployment costs. These technologies are particularly advantageous in rural areas, where traditional cellular networks may be sparse or nonexistent. Moreover, their low power consumption allows for extended battery life in devices, making them ideal for applications that require long-term monitoring without frequent maintenance. A comparison between the two protocols is highlighted in Table 2.1.

Table 2.1: Comparison between LoRa and SigFox [2]

Features	**LoRa**	**SigFox**
Technology	Semtech- Long range wireless network	Ultra Narrow Band (UNB)
Modulation	CSS (Chirp Spread Spectrum modulation)	DBPSK and GFSK
Bandwidth	8KHz to 1.6MHz	100 Hz
Maximum Data Rate	50Kbps	100bps
Bidirectional	Bi-directional in nature, half duplex, full duplex (FDX)	Limited/Half duplex
Maximum messages (per day)	Unlimited	140 (UL), 4 (DL)
Maximum payload (length)	243 bytes	12 bytes (UL), 8 bytes (DL)
Range	Very long range, more than 10 km in rural areas and 5 km in urban areas	30-50 km (Rural) 3-10 km (Urban)
Interference Immunity	Very high	Very high
Adaptive Data Rate	Allowed	Not Allowed
Standardization	LoRa-Alliance	Sigfox company is collaborating with ETSI on the standardization
Protocol	MAC layer protocol (LoRaWAN)	Lightweight protocol to handle small messages
Architecture	A star-of-stars topology	Star network

The major drawbacks of Sigfox are its limited data payload and limited bandwidth , which can restrict the types of applications it can support. In contrast, LoRa offers greater flexibility with larger payload sizes and the ability to handle more devices simultaneously, making it a more suitable choice for diverse IoT deployments. Thus we will be using the LoRa protocol for our project, as it aligns better with our requirements for scalability and data transmission needs.

2.2 LoRa

At the physical layer, Semtech's LoRa [14] employs a modulation technique that is a derivative of the Chirp Spread Spectrum (CSS). CSS enables the technology to facilitate extensive coverage while simultaneously providing robustness against interference, signal fading, and Doppler effects. This makes it an effective candidate to be used in urban environment. Additionally, Frequency Shift Keying (FSK) modulation can also be employed. Although FSK enables low-power communication, it does not attain the communication range afforded by the LoRa CSS. Typically, in scenarios where CSS modulation is implemented, the information-carrying signal, subsequent to undergoing a Forward Error Correction (FEC) process, is modulated into a sequence of chirp pulses prior to transmission. A chirp pulse is defined as a signal with a linearly increasing frequency that traverses the channel bandwidth (BW) centered around a specified central frequency (CF) depicted in Fig. . LoRa CSS signals operating within a particular BW may also exploit various spreading factors leading to different processing gains and spectral efficiency.

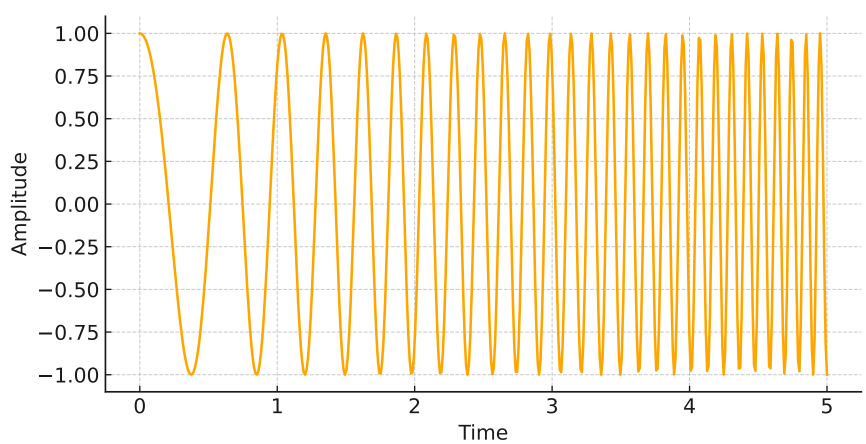

Figure 2.2: LoRa chirp pulse

LoRa offers a trade-off between sensitivity and data rate, while operating in a fixed-bandwidth channel of either 125 KHz or 500 KHz (for uplink channels), and 500 KHz (for downlink channels) [17]. Within this framework, each symbol is encoded utilizing a sequence of chips defined by a linearly swept frequency, commonly referred to as chirps. The duration of each chirp is determined by the signal's bandwidth, which subsequently establishes the chirping rate. A 500 kHz bandwidth is associated with a chirping rate of 125 kilo-chips-per-second (kcps) [15], or equivalently, 2 microseconds per chirp.

Within the same channel frequency for a given bandwidth (BW), each chirp (often called a sample) is differentiated from other chirps using an identical spreading factor (SF) based on the initial signal frequency. The rate of frequency increase depends on the spreading factor, which ranges from 6 to 12 [48]. Lower spreading factors cover the channel bandwidth faster than higher ones, leading to higher data transmission rates, but also increasing susceptibility to noise and interference, and reducing coverage. Thus, data can be encoded by combining the specified channel bandwidth with the chosen spreading factor, allowing for modulation over the central frequency of the selected channel. Table. 2.2 shows how the SF has an impact on range, bit-rate and Time on air for an 11-byte payload [17].

Table 2.2: LoRa Spreading Factor, Bit Rate, Range, and Time on Air for an 11-byte Payload

Spreading Factor (For UL at 125 KHz)	**Bit Rate**	**Range (Depends on Terrain)**	**Time on Air for an 11-byte payload**
SF10	980 bps	8 km	371 ms
SF9	1760 bps	6 km	185 ms
SF8	3125 bps	4 km	103 ms
SF7	5470 bps	2 km	61 ms

A chirp signal encompasses a number of SF bits, made possible by 2^{SF} potential chirp states within a specified chirp duration, thereby allowing LoRa to achieve a range of data transmission rates. The chirp duration, denoted as T_s, can be calculated using the following equation: The chirp duration, denoted as T_s, can be calculated using the following equation:

$$T_s = \frac{2^{SF}}{BW} \tag{2.1}$$

Additionally, factoring in the information from Forward Error Correction (FEC), known as the Coding Rate (CR), with typical values of $4/5$, $4/6$, $4/7$, and $4/8$, the bit rate, R_b, can be computed as:

$$R_b = \frac{SF \times BW}{2^{SF} \times CR} \tag{2.2}$$

A lower CR (e.g., $4/8$) improves transmission reliability in poor channel conditions, though at the cost of increased overall bit overhead compared to a higher CR (e.g., $4/5$). Moreover, higher SF values enable longer transmission distances due to increased receiver sensitivity, leading to more robust signal characteristics. However, these higher SF values also result in lower transmission rates and longer time on air (ToA) for packets of the same payload size. Furthermore, using larger bandwidth channels can boost transmission rates, but this typically reduces transmission distance. Transmission power can also be adjusted to extend the range and improve the likelihood of overcoming potential channel losses [22]. Concurrent uplink transmissions from multiple end devices (EDs) using the same channel but different SFs (i.e., exhibiting orthogonality) can coexist and be simultaneously decoded at a multi-channel gateway. Various SF and channel configurations can ensure successful simultaneous uplink transmissions. However, signals using different SFs are quasi-orthogonal, allowing overlapping signals to coexist as long as there is a minimum Signal-to-Interference-plus-Noise Ratio (SNR) between them. Multi-channel LoRa gateways can decode up to eight simultaneous quasi-orthogonal uplink transmissions. It is important to note that due to the half-duplex nature of both LoRa gateways and end nodes, simultaneous transmission and reception are not possible. Extensive studies have examined the Chirp Spread Spectrum (CSS) modulation technique, validating its properties such as resistance to interference and quasi-orthogonality [20], and proposing methods to achieve perfect orthogonality, better spectral efficiency, and improved Bit Error Rate (BER) performance [21].

LoRaWAN defines the MAC and link layer on top of the LoRa PHY layer, managed by the LoRa Alliance, and is widely used for IoT applications. However, while it provides robust features like authentication, encryption, and network management, it introduces significant overhead that can be detrimental to real-time and energy-sensitive applications. These overheads include complex authentication procedures, acknowledgment mechanisms, frame headers, and duty cycle regulations, all of which increase latency and energy consumption. These added layers can slow down both transmission and reception, making the protocol less suitable for applications requiring fast, low-power communication, such as wildlife tracking or industrial monitoring.

For this thesis, which aims to develop a highly efficient, low-latency, and scalable network, working directly on the LoRa PHY layer avoids these unnecessary overheads. By bypassing LoRaWAN, the system can use a custom MAC protocol tailored to the application's needs, reducing transmission delays, improving power efficiency, and allowing for more precise control of time-synchronized communication. This approach enables the creation of a lightweight, optimized solution ideal for large-scale networks where performance, low power usage, and real-time communication are critical.

2.3 Time-Synchronization

Time synchronization is one of the most important aspect for modern day WSLs. A common notion of time ensures that data is processed and logged correctly, which enables correlation of events across the system. It is required for many application in WSLs like scheduling, logging, and real-time analytics, where even minor discrepancies can lead to significant errors in data interpretation. It is inherently challenging to achieve and even more so in distributed systems where there is a lack of a shared memory [32]. Additionally, energy consumption also plays a crucial role on the network lifetime. The right protocol needs to strike a balance between energy efficiency and synchronization accuracy.

2.3.1 Clock Model

Every node in a WSN has an individual crystal oscillator based clock similar to any computing device. The frequency of the oscillator determines the rate at which the clock runs. The precision of these oscillators vary leading to drift over time due to variations in crystal imperfection, temperature, voltage, and other environmental factors.

Let us denote the value of the node clock at any given moment as $C(T)$, where T represents the time variable. In an ideal scenario $\frac{\partial C(t)}{\partial t}$ (clock skew) which is the derivative of the node clock value with respect to time, should equal 1, indicating that the clock is perfectly synchronized with the passage of real time. However, in practical applications, this ideal condition is rarely achieved. For instance, consider a frequency error as minimal as 0.001% [33]. This seemingly negligible discrepancy can lead to a significant deviation in clock accuracy, resulting in the clock being off by approximately 1 second every day. This illustrates how even if two nodes are initially synchronized to a common time reference, they can drift apart over time due to clock skew, which is the gradual divergence in the rate at which each clock ticks. It is also important to note that even in the absence of skew, clocks on different nodes may not share a common time reference.

This situation is characterized by what is known as a clock phase offset. The phase offset occurs when two clocks are set to different starting points, leading to a consistent difference in their readings despite the absence of drift. When we consider both clock skew and phase offset, the behavior of a node's clock can be comprehensively modeled. This model must account for the variations in both the rate of timekeeping (skew) and the initial alignment of the clocks (phase offset), providing a more accurate representation of how a node's clock will behave over time in a distributed system. The local clock of a node n_a can be approximated as:

$$C_a(t) = \alpha_a * t + \beta_a \tag{2.3}$$

where α_a represents the skew and β_a denotes the phase offset.

The relative equation between the clocks of two nodes, n_a and n_b, can then be expressed as:

$$C_a(t) - C_b(t) = (\alpha_a - \alpha_b) \cdot t + (\beta_a - \beta_b) \tag{2.4}$$

This relationship highlights how discrepancies in both skew and phase offset contribute to the overall time difference observed between the two nodes.

2.3.2 Decomposition of Delay

In order for WSNs to synchronize they need access to a communication channel that allow them to exchange timing information. However reliable the channel non deterministic delays in communication can further exacerbate the timing discrepancies caused by the inherent clock differences. The inherent delays in a channel according to [34] can be decomposed as:

- **Send time:** When a node opts to send a packet, it is queued as a task within a standard sensor node. Time is devoted to the packet's construction at the application layer before it is handed over to the lower layers for transmission. This period encompasses the delay experienced by the packet in reaching the MAC layer from the application layer. This delay is notably variable owing to the software delays caused by the operating system.

- **Access time:** This refers to the duration required for the packet to gain access to the communication medium, which can be influenced by contention among multiple nodes attempting to transmit simultaneously. It is entirely up to the MAC protocol and is variable.

- **Transmission time:** This is the time taken to push all the packet's bits onto the physical layer, determined by the packet size and the bandwidth of the communication channel. This delay is usually pretty deterministic and the software implementation of the transmission protocol can add additional delay.

- **Propagation time:** This is the time it takes for the packet to travel the wireless link from the sender to the receiver, which depends on the distance between them and the speed of the signal in the medium.

- **Reception time:** Once the packet reaches the receiver, this is the time required for the receiver to process the incoming signal and pass it to the MAC layer. This is usually deterministic in nature.

- **Receive time:** This includes the time taken to construct the bits into a packet, decode it in the application layer, check for errors, and prepare the data for further processing by the higher layers of the protocol stack. This process can vary based on the variable delays introduced by the operating system.

The authors in [42] described the message propagation in a wireless channel in more detail. They highlighted additional sources of delays:

- **Interrupt Handling Time:** This refers to the latency observed between the activation of the radio chip and the subsequent response from the microcontroller to an interrupt signal.

This duration is predominantly less than several microseconds (due to the requirement for the microcontroller to complete the currently executing instruction); however, in scenarios where interrupts are disabled, this latency can significantly increase.

- **Encoding Time:** This refers to the duration required for the radio chip to encode and convert a segment of the message into electromagnetic waves, commencing from the moment it triggers an interrupt signifying the reception of the signal from the microcontroller. This duration is deterministic and typically falls within the range of a hundred microseconds.

- **Decoding Time:** This refers to the duration required for the radio chip on the receiver's end to convert and interpret the message from electromagnetic waves into binary data. The process concludes when the radio chip generates an interrupt signifying the reception of the packet. This duration is predominantly deterministic and typically occurs within the range of hundreds of microseconds. Nevertheless, variations in signal strength and errors in bit synchronization may introduce jitter.

2.4 MAC Protocols

There are a multitude of MAC protocols that have been developed for low-power WSLs which are suitable of handling single-hop as well as multi-hop communication. Their aim is to ensuring reliable data transmission across the network while still optimizing energy consumption. As time synchronization is a key theme in this framework we are going to categorize the protocols based on their synchronization mechanisms. These categories include synchronous and asynchronous methods, each with distinct advantages and trade-offs in terms of energy efficiency and complexity.

2.4.1 Asynchronous MAC Protocols

These protocols operate without a global time reference, allowing nodes to communicate independently. Usually, a contention based mechanism is employed to give channel access to competing nodes within a network. These protocols have a high degree of adaptability to changing network topology but this comes at a cost [29]. The overhead associated with these mechanisms can lead to several challenges, including increased instances of overhearing—where nodes unintentionally receive signals not intended for them—frame collisions, which occur when multiple nodes attempt to transmit simultaneously, and channel-idle listening, where nodes consume resources while waiting for an opportunity to transmit. These issues can significantly hamper a network's overall performance, leading to increased latency and reduced throughput. Thus, while contention based MAC protocols provide flexibility in a changing network environment, they also introduce complexities that must be carefully managed to optimize communication efficiency.

B-MAC (Berkley MAC) [25] is one such protocol designed based on this approach, allowing nodes to sleep and wake based on the presence of activity in the channel. It supports Carrier Sense Multiple Access (CSMA) along with low-power listening (LPL) where the nodes periodically wake up to check for incoming transmissions, thereby conserving energy while still maintaining the ability to respond promptly to network events. A long wake-up preamble is sent before the actual data transmission, which helps in synchronizing the nodes and ensuring that they are awake during critical communication periods. This is leads to peripheral nodes to sample the channel extremely frequently even when the activity interval spans over several minutes. This leads to reduced energy efficiency and coupling it with listening for length preambles severely restricts the scalability of the system. X-MAC [28] is designed to address these challenges by implementing a more efficient preamble structure allowing receivers to respond during preamble transmission minimizing energy consumption while still maintaining features like LPL and sleep scheduling. But it is still plagued by two major drawbacks and that is the unavailability of the peripheral' node's exact wake-up time and an inability to deal with collisions [23]. T-MAC [27] (Timeout MAC) is another approach that introduces a timeout mechanism to improve energy efficiency. By dynamically adjusting the active and sleep periods based on traffic conditions, T-MAC aims to reduce idle listening while still ensuring timely data delivery. However, it also faces challenges in scenarios with bursty traffic, where nodes may miss critical transmissions due to extended or early sleep intervals. S-MAC [26] is another protocol designed to enhance energy conservation by using periodic sleep schedules and message passing to synchronize nodes. While S-MAC effectively reduces energy consumption in low-traffic environments, it struggles with scalability and can lead to increased latency in high-density networks.

2.4.2 Synchronous MAC Protocols

These protocols are designed to guarantee a shared medium access to all nodes, ensuring that transmissions occur in a coordinated manner. They essentially operate on a TDMA scheme facilitating a highly organized method of data transmission, allowing multiple nodes to communicate without interference. This approach minimizes collisions, preserves energy and maximizes throughput, but it requires precise time synchronization among nodes. This can be particularly challenging as time slots are small clocks are prone to drifting in peripheral nodes. To facilitate the TDMA operation, the wireless communication channel is segmented into structured units known as superframes. Each superframe is further divided into specific timeslots that are allocated to peripheral nodes, granting them exclusive periods during which they can transmit their data. This meticulous organization not only enhances the reliability of the communication process but also ensures that all nodes can efficiently share the medium without the risk of interference, thus maintaining the integrity and performance of the network as a whole.

TRAMA [24] is Synchronous, collision -free TDMA based protocol that optimizes bandwidth utilization by dynamically adjusting the timeslot allocation based on the traffic demands of each node, thereby improving overall system efficiency. When nodes are not participating, they switch to a low-power state that conserves energy and prolongs the operational lifespan of the network devices. As the timeslots are dynamically adjusted, deterministic latency cannot be expected for message exchanges which can lead to variability in response times, potentially impacting time-sensitive applications that require consistent delivery intervals. FLAMA (Flow-Aware MAC) [30] improves over TRAMA by preventing data to be transmitted to nodes not ready to recieve data leading to greater efficiency in data handling and reducing unnecessary retransmissions, thus enhancing the overall throughput of the network. It also improves on energy efficiency and network latency over TRAMA but still has limitations in scalability and scenarios with highly dynamic topologies, where the flow-aware mechanisms may struggle to adapt quickly enough to changing conditions. Then there is Z-MAC (Zebra MAC) which tries to strike a balance by combining the benefits of both contention-based and scheduled access methods, allowing for flexibility in various network conditions while maintaining a level of fairness among nodes. Under hight traffic conditions it operates on a TDMA scheme and when the network traffic is low, switched to CSMA. This leads to improved performance over B-MAC during high traffic loads but that’s not the case during low traffic periods, where the overhead of switching between modes can introduce latency and reduce overall efficiency.

All of these protocols take global time synchronization for granted and don’t account for any steps to achieve it. There is a need for time synchronization to be a core tenet of a scheduling based MAC protocol to ensure that all nodes can effectively coordinate their transmissions and minimize collisions, ultimately enhancing the reliability and throughput of the network. That is what this work has set out to achieve.

Chapter 3

Design and Implementation

3.1 System Architecture

The design and implementation of the modular framework for Wireless Sensor Networks (WSNs) is based on leveraging a LoRa-based low-power hardware platform integrated with an efficient TDMA-based medium access control (MAC) protocol. This section provides a detailed overview of the system architecture, highlighting key components and their interactions.

3.1.1 Overview of the System Architecture

The system architecture of the modular framework is designed to address critical challenges in WSNs, such as time synchronization, scheduling, and energy efficiency. The architecture follows a star topology, where a Coordinator node, equipped with a LoRa transceiver, acts as the synchronization hub and coordinator for peripheral nodes. The star topology is a crucial component that simplifies the complexity of network design and routing protocols. The main components of the system architecture are:

- **Coordinator Node:** The Coordinator node is responsible for managing network communication, sending synchronization packets, and TDMA slot assignment. It integrates a LoRa transceiver, a microcontroller unit (MCU), and a GPS module (optional) for precise time synchronization.

- **Peripheral Nodes (Motes):** Peripheral nodes are sensor nodes equipped with LoRa transceivers and MCUs. These nodes collect data and communicate with the Coordinator node. They receive synchronization packets and slotting information and adjust their internal clocks accordingly to maintain precise timing.

- **Synchronization Mechanism:** The Coordinator node periodically sends synchronization packets containing the current timestamp. Peripheral nodes use these packets to synchronize their clocks, ensuring coordinated operations.

- **Slotting and Scheduling:** The MAC protocol implements a TDMA-like slotting mechanism, where each peripheral node is assigned specific time slots for data transmission. This reduces collisions, ensures predictable throughput and leads to better channel utilization.

- **Power Management:** The system architecture incorporates energy-efficient techniques, such as low-power modes and dynamic power adjustment, to extend the battery life of sensor nodes.

3.1.2 Coordinator Node

The Coordinator node is the core component of the modular framework, responsible for network management and synchronization. Key features of the Coordinator node include:

- **LoRa Transceiver:** Provides long-range communication with peripheral nodes, leveraging LoRa's robust and low-power capabilities.

- **Microcontroller Unit (MCU):** Manages synchronization, scheduling, and data processing tasks. The MCU interfaces with the LoRa transceiver and GPS module.

- **GPS Module:** Provides precise time information, which can be added to synchronization packets. The GPS time lock ensures high accuracy in time synchronization.
- **Synchronization Packet Transmission:** The Coordinator node periodically transmits synchronization packets containing the current GPS timestamp. The peripheral nodes receive these packets and adjust their clocks accordingly.
- **Invitation Packet Transmission:** The Coordinator node transmits invitation packets containing slotting information in response to on-boarding requests from the peripheral nodes.

3.1.3 Peripheral Nodes

Peripheral nodes, also known as motes, are responsible for data collection and communication with the Coordinator node. Key features of peripheral nodes include:

- **LoRa Transceiver:** Enables long-range communication with the Coordinator node, ensuring reliable data transmission over large areas.
- **Microcontroller Unit (MCU):** Manages data collection, synchronization, and communication tasks. The MCU interfaces with the LoRa transceiver and sensors.
- **Sensor Interfaces:** Peripheral nodes can be equipped with various sensors to monitor environmental parameters such as temperature, humidity, and light. The collected data is processed and transmitted to the Coordinator node.
- **Clock Adjustment:** Upon receiving synchronization packets from the Coordinator node, peripheral nodes adjust their clocks to match the received timestamp. This ensures precise time-synchronization across the network.
- **Power Management:** Peripheral nodes implement energy-efficient techniques, such as low-power modes and adaptive power control, to conserve battery life.

3.1.4 Communication and Data Flow

The communication and data flow in the modular framework are designed to ensure efficient and reliable network operation. The key steps in the communication process are the following:

1. **Initialization:** The Coordinator node initializes the network by broadcasting setup information, including slot assignments and synchronization schedules.

2. **Synchronization:** The Coordinator node periodically sends synchronization packets to all peripheral nodes. Nodes adjust their clocks based on the received timestamp, ensuring precise time-synchronization.

3. **Data Collection:** Peripheral nodes collect data using their sensors. The data is processed and stored locally until the assigned transmission slot.

4. **Data Transmission:** During their assigned time slots, peripheral nodes transmit the collected data to the Coordinator node. The Coordinator node coordinates the schedule to minimize collisions and optimize channel utilization.

5. **Data Aggregation and Processing:** The Coordinator node aggregates the data received from peripheral nodes and processes it for further analysis. The processed data can be forwarded to a Coordinator server or cloud for storage and analysis.

6. **Power Management:** Nodes enter low-power mode with their radios turned off when not transmitting or receiving, conserving energy.

3.2 Time-Synchronization

Time-synchronization is a fundamental requirement in Wireless Sensor Networks (WSNs) for coordinated operations, precise data aggregation, and efficient resource management. The modular framework presented in this research employs a robust time-synchronization method using synchronization packets, with an optional GPS time lock feature for applications requiring real-world timing.

3.2.1 Core Synchronization Method Using Synchronization Packets

The primary method for achieving time-synchronization in the modular framework involves the use of synchronization packets broadcasted by a Coordinator node. This method ensures that all peripheral nodes in the network operate in a common time frame, which is crucial for coordinated data transmission and network operations.The steps involved with the implementation of the core synchronization are as follows:

1. **Coordinator Node Timekeeping:** The Coordinator node maintains an accurate internal clock, which serves as the reference time for the network. This internal clock can be disciplined by the GPS module's pulse per second (PPS) signal as well to counteract drift and maintain timing precision.

2. **Synchronization (Sync) Packet Transmission:** The Coordinator node periodically broadcasts synchronization packets containing the current timestamp to all peripheral nodes. These packets are sent at regular intervals, ensuring consistent synchronization across the network.

3. **Clock Adjustment at Peripheral Nodes:** Upon receiving the synchronization packets, peripheral nodes adjust their local clocks to match the reception of the SYNC packet. This process aligns the clocks of all nodes, creating a synchronized network environment.

This leads to numerous advantages, such as simplicity and efficiency, as the method is straightforward to implement and requires minimal computational resources, making it energy-efficient. It also supports scalability, with synchronization packets capable of being broadcast to a large number of nodes, making the system suitable for extensive networks. Additionally, the method provides adaptability, offering a flexible foundation that can be tailored to various WSN applications without the need for additional infrastructure.

3.2.2 Optional GPS Time Lock Feature

The GPS time lock is an optional feature designed for applications that require synchronization with real-world timing, such as timestamping data with actual date and time. This feature enables the protocol to transmit unix time as part of the Sync packet, ensuring that the network's internal time aligns with global time standards. It is implemented as follows:

1. **GPS Module Integration:** The Coordinator node is equipped with a GPS module that receives precise timing information from GPS satellites.

2. **PPS Signal Utilization:** The Coordinator node utilizes the Pulse Per Second (PPS) signal provided by the GPS module to discipline its internal clock. This prevents clock drift at the Coordinator node, maintaining high synchronization accuracy.

3. **Broadcasting Synchronization Packets:** The Coordinator node continues to broadcast synchronization packets, now incorporating the disciplined GPS-based timestamp. Peripheral nodes adjust their clocks based on these packets, aligning with the GPS time.

The GPS Time Lock feature provides synchronization with global time, which is essential for applications requiring precise timestamping and time-based data analysis. It offers high accuracy by utilizing the Pulse Per Second (PPS) signal from the GPS module, ensuring that the Coordinator node's clock maintains high precision over time, thereby reducing drift and enhancing the reliability of time synchronization across the network.

3.2.3 Implementation and Integration

The combined use of synchronization packets, optional GPS time lock provides a comprehensive solution for time-synchronization in the modular framework. This approach ensures that all nodes in the network are synchronized to a common time frame, whether or not real-world timing is required. It is implemented as follows:

1. **Initial Setup:** The Coordinator node sets up the network, establishing its internal clock or disciplining it with GPS if the time lock feature is used.

2. **Regular Synchronization:** The Coordinator node broadcasts synchronization packets at regular intervals, maintaining a synchronized network.

3. **Optional GPS Time Lock Activation:** For applications requiring real-world timing, the GPS module provides precise timing, which is incorporated into the synchronization packets.

3.2.4 System Decomposition of Delay

To achieve a more accurate and effective modeling of the synchronization system, it is crucial to conduct a thorough evaluation of the inherent delays present in the synchronization process. As illustrated in Fig. 3.1, the synchronization procedure involves the transmission of a Sync packet by the coordinator, which serves as a reference point for the nodes to synchronize their clocks upon its reception. However, it is essential to note that the interrupt signal generated by the LoRa radio signifies the completion of a transmission (Tx) or reception (Rx) event, yet it does not provide an exact timestamp for when a packet is either sent or received. This inherent lack of precision can lead to timing discrepancies, potentially undermining the overall performance and reliability of the system.

Delving deeper into the synchronization process, the initiation of the Superframe occurs at the rising edge of the Pulse Per Second (PPS) signal. The time taken for packet construction at the application layer, denoted as T_{Send}, is a critical factor, as it represents the duration required before the packet is passed down to the lower layers for transmission. Following this, T_{Encode} refers to the time necessary for encoding the packet into a format that is suitable for wireless transmission via the radio. In the specific hardware configuration being analyzed, it was observed that the combined duration of $T_{Send} + T_{Encode}$ averages around 10 milliseconds, with a variability of approximately ±10 microseconds.

Next, we consider $T_{Transmission}$, which is equivalent to T_{ToA} (Time of Arrival) for a LoRa packet. This can be determined based on several parameters, including the packet size, spreading factor (SF), bandwidth, and coding rate [15]. Additionally, T_{Ramp} signifies the time delay that exists between the Tx Done Interrupt and the actual commencement of transmission, which is an important factor to consider in the overall timing analysis.

On the receiving end, T_{Prop} denotes the propagation delay, which is contingent upon the physical distance separating the transmitter and the receiver. Furthermore, $T_{Reception\_Time}$ represents the duration required for the receiver to process the incoming signal and extract the pertinent data from it. For the system under discussion, the relationship holds that $T_{ToA} = T_{Prop} + T_{Reception\_Time}$.

Following this, we have T_{Decode}, which accounts for the time taken to decode the received signal and convert it back into a usable format, subsequently raising an interrupt to indicate the successful reception of the packet. In this particular setup, the delay associated with decoding has been measured to be approximately 500 microseconds, with a variability of ±10 microseconds.

Ultimately, the total latency of the communication system can be succinctly expressed by the equation $T_{Total} = T_{Send} + T_{Encode} + T_{ToA} + T_{Decode}$. These delay values are highlighted in Table. 3.1.

Table 3.1: Summary of network delay values

Measurement	Value
T_{ramp}	43µs ± 3µs
$T_{Encode} + T_{Send}$	10ms ± 5µs
T_{decode}	500µs ± 10µs
T_{prop}	1µs per 300m of range
$T_{Reception\_Time}$	$T_{ToA} - T_{prop}$
$T_{Transmission\_Time}$	T_{ToA} [16]

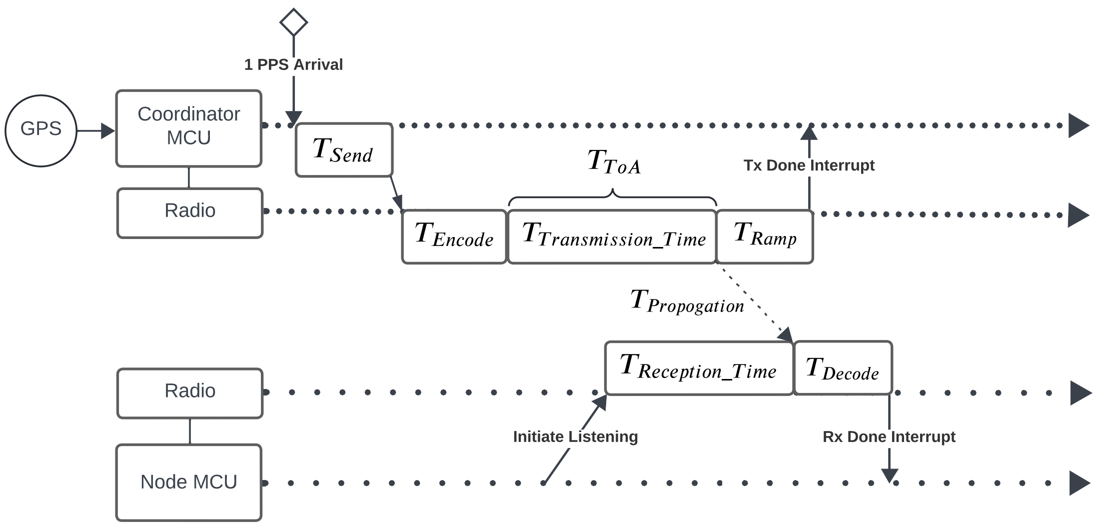

Figure 3.1: Decomposition of Delay

3.3 Scheduling Protocol

In the context of Wireless Sensor Networks (WSNs), slotting and scheduling are critical for managing medium access and ensuring efficient channel utilization. The modular framework presented here employs a sophisticated slotting mechanism inspired by Time Division Multiple Access (TDMA), which divides the communication timeline into discrete time slots. This approach is vital to achieve reliable throughput and minimize interference between nodes. The slotting mechanism within the modular framework is foundational for orchestrating network communications.

3.3.1 MAC Protocol Implementation

The LoRa-based MAC protocol operates with two main device categories: the coordinator and the Nodes. Nodes are initially unscheduled (referred to as UN), with no synchronization or slotting information. The coordinator plays a pivotal role in this system, periodically transmitting synchronization packets and slotting information, transforming UNs into scheduled nodes (SN).

Superframe Structure and Operations

The coordinator operates on a 1-second cycle, known as the LoRa Superframe (SF) shown in Fig. 3.2, which consists of three main components: SYNC packet transmission, Contention Access Period (CAP), and Invitation period. The sequence of operations within the SF is described as follows.

1. **SYNC Packet Transmission:** The SF begins with the transmission of the SYNC packet, which typically takes around 30ms. However, to ensure reliability, 50ms is allocated for this process. The SYNC packet serves as a time reference for all nodes, enabling synchronization. A correction packet can also be transmitted during this time to change the slotting information for a SN.

2. **Contention Access Period (CAP):** Following the SYNC packet, the coordinator enters a listening mode for the CAP, lasting 750ms. During this period, UNs transmit their node IDs after a random back-off interval ranging from 10 to 700ms. These transmissions, known as Announcement packets, are used by the coordinator to identify and manage the nodes within the network.

3. **Invitation Period:** At the end of the CAP, the coordinator transmits an Invitation packet containing the node IDs received during the CAP and the corresponding slot assignments. This period typically requires around 155-170ms for processing, with 200ms allocated to ensure reliability.

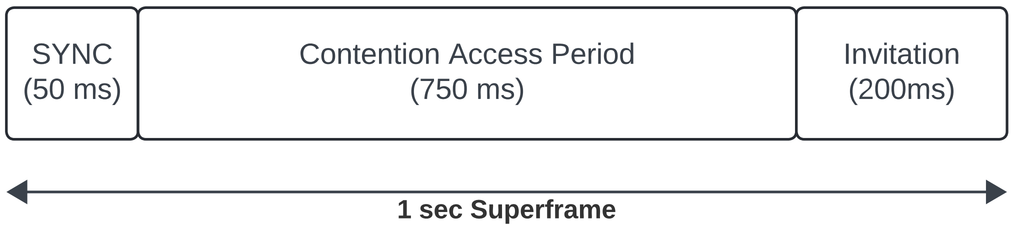

Figure 3.2: LoRa Superframe structure

The 1-second SF is thus composed of 50ms for SYNC packet transmission, 750ms for the CAP, and 200ms for Invitation packet transmission. The SF can be extended to accommodate more Announcement packets, but this extension may delay the onboarding of new nodes that missed the initial SYNC packet. After rigorous testing, a 1-second SF was found to provide a good balance between node serviceability and operational reliability. Details about the packets are mentioned in Table 3.2. The state diagram for coordinator operation is shown in Fig. 3.3.

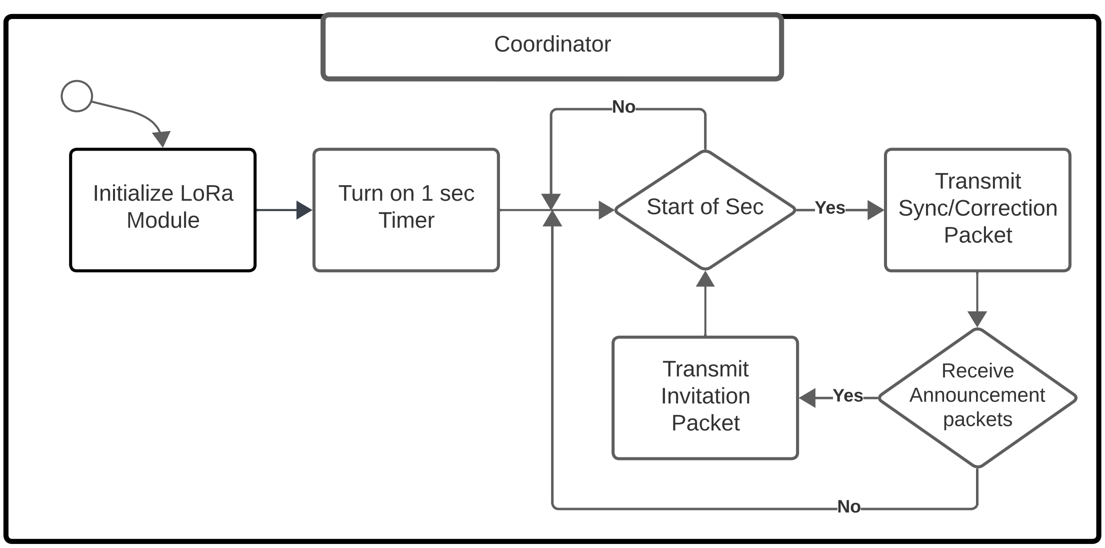

Figure 3.3: coordinator States Flowchart

3.3.2 Node operation States

There are currently three distinct states for the node operation as shown in Fig. 3.4: Initialization, Discovery, and On-boarded which operate as follows:

1. **Initialization**: At startup, the LoRa module is initialized, and the unscheduled node is switched to Discovery mode with listening enabled.

2. **Discovery**: The node waits to receive a SYNC packet. Upon receiving a SYNC packet from the coordinator, the node transmits an Announcement packet (containing the node ID) with a random delay of less than 700 milliseconds to avoid collisions with other tags. This Announcement packet is received by the coordinator during the Contention Access Period (CAP) of the Superframe (SF). The node then switches to listening mode to receive the Invitation Packet. Based on the Announcement packets received, the coordinator transmits an Invitation packet at the end of the SF. If the Invitation packet contains slotting information (based on the node ID), the node switches to the On-boarded state, becoming a scheduled node.

3. **On-boarded**: In this state, the node has the slotting information but still needs to calibrate its clock with the coordinator. Listening is turned on, and at the arrival of the next SYNC packet, the node's clock is synchronized to its reception. This synchronization ensures that all nodes have a common time reference, which is essential for the MAC protocol to function correctly.

Addressing erroneous tags: As a safety measure, we created a correction method to reassign colliding tags to different slots. Time synchronization failures at the tags may result in transmissions at incorrect time slots, causing persistent collisions among tag groups. By monitoring the tags experiencing frequent collisions, the coordinator sends a correction packet via LoRa to reassign the incorrect tags. The correction packet is part of the invitation packet.

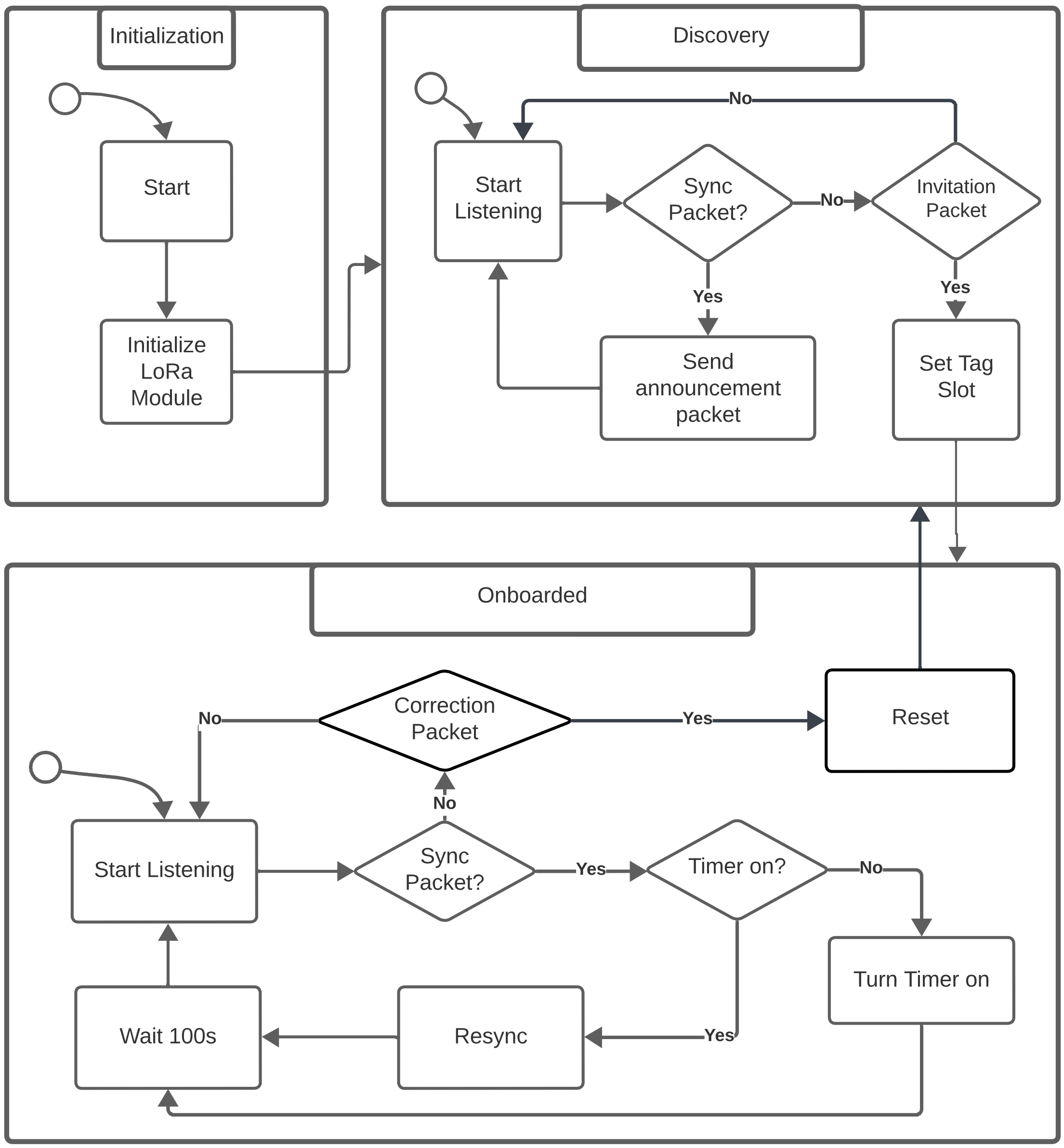

Figure 3.4: Node States Flowchart

3.3.3 Clock Drift Management

Clock drift is a common issue in WSNs due to hardware variations and environmental factors. To manage clock drift, a Re-Sync protocol is implemented. Member nodes (SNs) turn on

Table 3.2: Packet characteristics for various packet types.

Packet Characteristic	Payload (Bytes)	Preamble (Symbols)	Explicit Header	CRC
SYNC Packet	1	10.25	Disabled	Disabled
Announcement Packet	2	10.25	Disabled	Disabled
Invitation Packet	11	10.25	Disabled	Disabled

their listening mode every 60 seconds to receive a SYNC packet and realign their clocks with the coordinator. This regular synchronization helps maintain clock accuracy and reduces the impact of drift.

During testing, clock drift among nodes ranged between 10-15 ppm relative to the coordinator, even without employing a clock drift compensation algorithm. The 60-second Re-Sync interval was chosen to keep drift within acceptable limits, ensuring reliable network synchronization without excessive energy consumption.

3.4 Versatile MAC

The modular framework's MAC protocol is designed with versatility in mind, enabling it to accommodate a wide range of applications and network configurations. This section explores additional features of the MAC protocol that enhance network functionality, including side channeling and integration capabilities.

3.4.1 Side Channeling

Side channeling is an innovative feature integrated into the MAC protocol that allows for secondary communication paths within the network. This capability is particularly useful for transmitting control messages or non-critical data alongside the primary communication streams.

Key Aspects of Side Channeling:

- **Dual-Purpose Channels:** The protocol designates specific channels or time slots for side channeling, ensuring that they do not interfere with the primary data transmission paths. This separation helps maintain the integrity and reliability of critical data communications.
- **Use Cases:** Side channels can be used for various purposes, such as:
  - **Control and Command Signals:** Transmitting network control commands, configuration updates, or synchronization messages.
  - **Health Monitoring:** Sending node status updates, battery levels, or other diagnostic information.
  - **Secondary Data Streams:** Transmitting less time-sensitive data that does not require the same level of prioritization as primary data streams.

3.4.2 Integration Capabilities with Other Systems

The MAC protocol's design emphasizes seamless integration with other network systems and technologies, making it adaptable for various WSN applications.

Integration Features:

- **Modularity:** The protocol's modular design allows for easy customization and integration with different types of sensors, actuators, and data processing systems. This modularity supports a wide range of application scenarios, from environmental monitoring to industrial automation.

- **Interoperability:** The protocol supports operation with other communication technologies, such as Bluetooth, Wi-Fi, UWB, and cellular networks. This interoperability is facilitated by using standardized communication interfaces and protocols, enabling the WSN to interact with external systems and broader IoT ecosystems.

- **Data Aggregation and Fusion:** The protocol enables data aggregation and fusion from multiple sources, allowing for comprehensive data analysis and decision-making. This capability is crucial in applications like smart cities, where data from various sensors must be integrated and analyzed in real-time.

- **Scalability and Flexibility:** The protocol's scalable architecture allows it to support networks of varying sizes, from small deployments with a few nodes to large-scale networks with hundreds or thousands of nodes. This scalability is achieved through efficient slotting and scheduling mechanisms, as well as dynamic resource allocation strategies. Flexibility in configuration allows network administrators to adjust protocol parameters, such as duty cycles, slot allocations, and transmission power settings, to optimize performance for specific application needs and environmental conditions.

3.5 Power Efficiency

In Wireless Sensor Networks (WSNs), optimizing power efficiency is crucial due to the limited energy resources of sensor nodes. Effective power management strategies are essential for extending the network's operational life, reducing maintenance costs, and ensuring continuous network performance. This modular framework focuses on several key methods for enhancing power efficiency, particularly Low-Power Listening (LPL) and duty cycling

3.5.1 Low-Power Listening (LPL)

Low-Power Listening (LPL) is a power-saving technique designed to reduce the time sensor nodes spend in active listening mode. By incorporating LPL into the MAC protocol, the framework ensures that nodes only activate their receivers during specific time windows when communication is expected, reducing unnecessary power consumption.

Implementation of LPL:

- **Scheduled Wake-Up:** Nodes are programmed to wake up briefly based on the Re-Sync period for synchronization (SYNC) or data packets. These wake-up periods are synchronized with the Coordinator's broadcast schedule, ensuring that nodes use active listening time efficiently.
- **Energy Savings:** LPL minimizes the duration and frequency of active listening periods, significantly reducing each node's power consumption and extending battery life.

3.5.2 Duty Cycling

Duty cycling is a crucial power management technique that alternates nodes between active and sleep modes. By limiting the time nodes spend in power-intensive states, duty cycling conserves energy while maintaining network functionality.

Key Aspects of Duty Cycling:

- **Active and Sleep Phases:** Nodes operate in active mode during Initialization and Discovery state. Once on-boarded, nodes enter a low-power sleep mode to conserve energy until their scheduled transmission slots.
- **Optimizing the Duty Cycle:** The duration of active and sleep phases is calibrated based on the network's communication needs and the energy profile of each node. The aim is to balance energy savings with the network's performance requirements. This is modular and can be adjusted.

Duty Cycle Configuration: The configuration of the duty cycle is adjusted according to the network's operational needs. For instance, in environments with infrequent data transmission, nodes can be configured to stay in sleep mode longer, waking up only when necessary for communication. This approach significantly reduces the overall energy consumption across the network.

Benefits of Duty Cycling:

- **Extended Node Lifespan:** By maximizing the time nodes spend in low-power sleep states, Duty Cycling significantly extends the operational lifespan of the sensor nodes. This is particularly valuable in scenarios where replacing or recharging batteries is difficult.
- **Efficient Resource Utilization:** Duty Cycling ensures that nodes utilize energy resources efficiently by staying active only during critical communication or network management periods. This leads to better energy conservation and improved network longevity.
- **Enhanced Data Integrity:** By reducing the active time of nodes, Duty Cycling minimizes the chances of data collisions and packet loss, thereby enhancing the overall integrity of the transmitted data.

Chapter 4

Experimental Methods & Result

4.1 Experimental Setup

To thoroughly evaluate the modular framework, a series of experiments were conducted. The setup included multiple peripheral nodes and a central coordinator node, configured to mimic real-world deployment scenarios.

The basic building blocks for the peripheral nodes and the Coordinator include a LoRa transceiver and a Microcontroller Unit (MCU). To achieve this, we chose to utilize the SX1272MB2xAS [9] mbed shield, which incorporates SX1272 transceivers [10]. These transceivers are equipped with the LoRa long-range modem, offering extensive spread-spectrum communication, exceptional interference resistance, and reduced power consumption.

The employed microcontroller board is the NUCLEO-L073RZ [11], featuring an ARM 32-bit Cortex-M0+ processor capable of operating at a maximum CPU frequency of 32MHz. The NUCLEO-L073RZ board comes with compatible headers, allowing the LoRa shield to be directly connected to the board without any retrofitting. This compatibility simplifies the integration process, making it easier to prototype and develop the communication system efficiently.

4.1.1 Testbed Configuration

- **Peripheral Nodes:** Multiple nodes equipped with LoRa transceivers and microcontrollers, configured to operate as both unscheduled nodes (UN) and scheduled nodes (SN).
- **Coordinator Node:** A central node that acts as the network coordinator, equipped with a LoRa transceiver, microcontroller, and a GPS module.
- **Network Topology:** A star topology was employed, with the Coordinator at the center and peripheral nodes distributed around it.

4.1.2 Metrics Evaluated

The experiments focused on the following key performance metrics:

- **Synchronization Accuracy:** The precision of the time synchronization between the central Coordinator and the peripheral nodes.
- **Throughput:** The rate at which data is successfully transmitted and received within the network.
- **Energy Efficiency:** The power consumption of nodes during various operational states, including synchronization, data transmission, and idle periods.
- **Network Reliability:** The consistency and robustness of communication under different network conditions and configurations.

4.2 Synchronization Accuracy

The accuracy of time synchronization was assessed through a meticulous comparison of the clocks of peripheral nodes against the clock of the central coordinator. This comparison was crucial in evaluating the precise alignment of time between the different components of the system.

The experiments delved into two critical aspects of synchronization: the initial synchronization process and the continued synchronization as time progressed. Through these experiments, the aim was to gain a comprehensive understanding of how the synchronization mechanism performed under varying conditions and over an extended period.

The experiments were structured into two distinct parts to capture different scenarios. One segment involved the coordinator node and the peripheral nodes being in close proximity to each other, allowing for a more controlled environment to study synchronization behavior. The second part, on the other hand, introduced a considerable separation of 20 meters between the coordinator node and the peripheral nodes. This significant distance aimed to simulate real-world scenarios where components are spread apart, testing the robustness of synchronization in challenging conditions. One of the primary reasons for choosing LoRa was its range and efficiency in transmitting data over long distances, which was crucial for maintaining communication between the coordinator node and peripheral nodes even when they were 20 meters apart.

The experimental setup devised for this rigorous time synchronization testing comprised two peripheral nodes and the central coordinator node. These components played a crucial role in the synchronization process, with the nodes specifically programmed to generate a 1Hz clock aligned precisely with the reception of the synchronization packet. The start of the clock is marked by a pulse at pin PC4 of the Nucleo board which was continuously monitored, enabling us to track any potential drift or inaccuracies over an extended observation period. To collect and analyze data on synchronization performance, a logical analyzer was employed. This analyzer was capable of sampling the clock at a high frequency of 16 MHz, providing detailed insights into the behavior of the synchronization process. The data gathered from the logical analyzer were thoroughly analyzed to evaluate the system's synchronization performance under varying distances between the coordinator node and peripheral nodes.

4.2.1 Testing Methodology

1. Initial Synchronization: Peripheral nodes were synchronized using the synchronization packets transmitted by the coordinator node. The characteristics of that initial synchronization were evaluated.

2. Ongoing Synchronization: As the unscheduled nodes are scheduled, the system's ability to maintain accurate time alignment over a period was also assessed.

4.2.2 Evaluation Metrics

We establish our evaluation criteria by assessing the precision of aligning our device clocks with GPS time and the efficiencies in sustaining this alignment. The subsequent indicators are employed for evaluating the effectiveness.

1. **Synchronization error**: This is the offset of the device clock from the GPS time reference, measured in microseconds, which indicates the level of accuracy in time synchronization achieved by the system. This quantification allows us to analyze the system's performance in maintaining time alignment under varying conditions and work towards optimizing synchronization processes for enhanced efficiency.

2. **Skew Rate**: Clock skew is the result of variations in the frequencies of clocks in separate sensor nodes, causing their time readings to slowly drift apart. This effect mainly arises from the natural inaccuracies present in the oscillators within the sensor nodes, which can

be affected by external factors like temperature and voltage changes. Skew rate gives us a rate of change in the clock's offset value over time, helping us characterize the dynamic nature of clock synchronization in WSNs.

For both of these, the timing of the coordinator node was evaluated with the internal clock disciplined by a GPS. The GPS-disciplined clock ensured precise timekeeping, allowing for a detailed examination of the robustness of the synchronization process in real-world scenarios.

The testing methodology was applied to both the close proximity and far-field scenarios to evaluate the system's performance under varying distances between the coordinator node and the end devices.

4.2.3 Evaluation of Initial Synchronization Performance

Synchronization with the coordinator node was measured and compared across multiple trials to assess the robustness of the system. For this, the Re-Sync period for the nodes was set to 1 second. The system was run for 200 seconds, leading to 100 re-Syncs and data from 3 trials for Node 1 was compiled and displayed in Table 4.1 for near-field and table 4.2 for far-field:

Table 4.1: Near Field Synchronization error (μs) over 3 trials

Trial	Mean	Std. Dev.	95% Confidence Interval
1	16.74	18.36	16.74 ± 19.29
2	17.65	20.58	17.65 ± 20.51
3	17.20	17.8	17.20 ± 18.68

The data highlights that the average synchronization accuracy is around 17±25µs for the near-field and the far-field cases. The negligible difference is due to the fact that propogation delay of 1us is added every 300m which is too small for our analysis This level of accuracy allows us to efficiently segment the 1-second superframe into 1,000 distinct time slots, each lasting 1 millisecond (ms). This precise division facilitates the seamless integration of up to 1,000

Table 4.2: Far Field Synchronization error (μs) over 3 trials

Trial	Mean	Std. Dev.	95% Confidence Interval
1	17.74	18.45	17.74 ± 19.45
2	17.59	19.21	17.59 ± 19.99
3	17.80	19.1	17.80 ± 19.21

peripheral nodes into the system. Such scalability is not merely advantageous; it is essential for applications that depend on extensive sensor networks, where meticulous timing and coordination are paramount for achieving optimal performance and ensuring the integrity of the data collected. To further elucidate the system's performance, we have plotted synchronization error against the Re-Sync number, providing a visual representation of our findings. Figure 4.1 illustrates the initial synchronization performance observed in the near-field scenario, while Figure 4.2 captures the corresponding data for the far-field case. Every point on the graph denotes the clock offset measured right after a resynchronization period. These visual aids serve to enhance our understanding of the synchronization dynamics and the overall efficacy of the system under varying conditions.

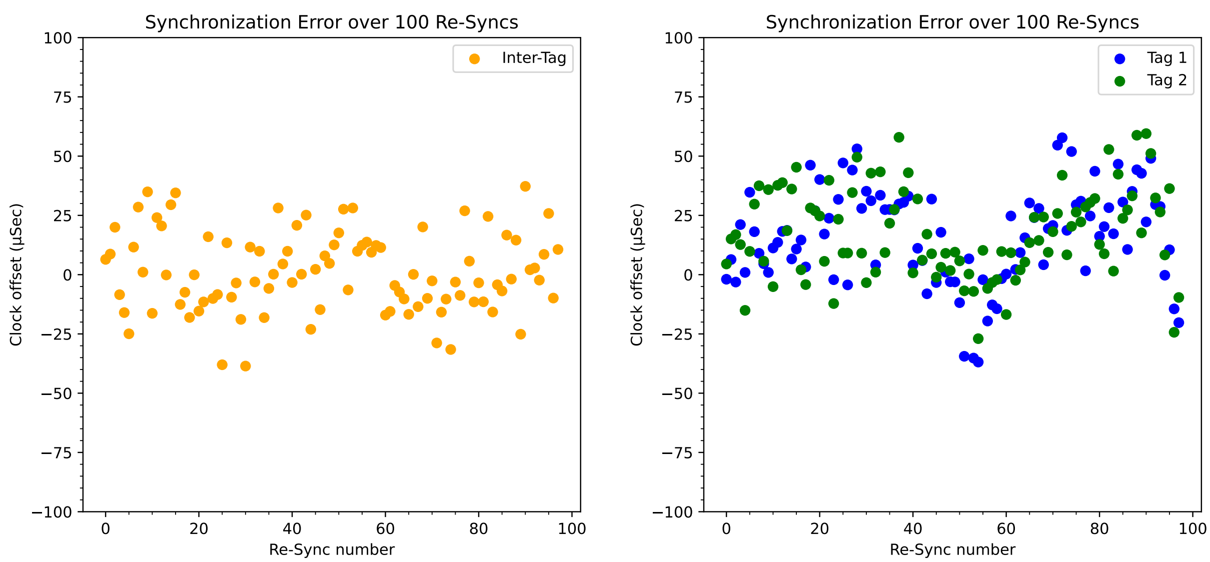

Figure 4.1: Synchronization Error over 100 Re-syncs [Near-Field]

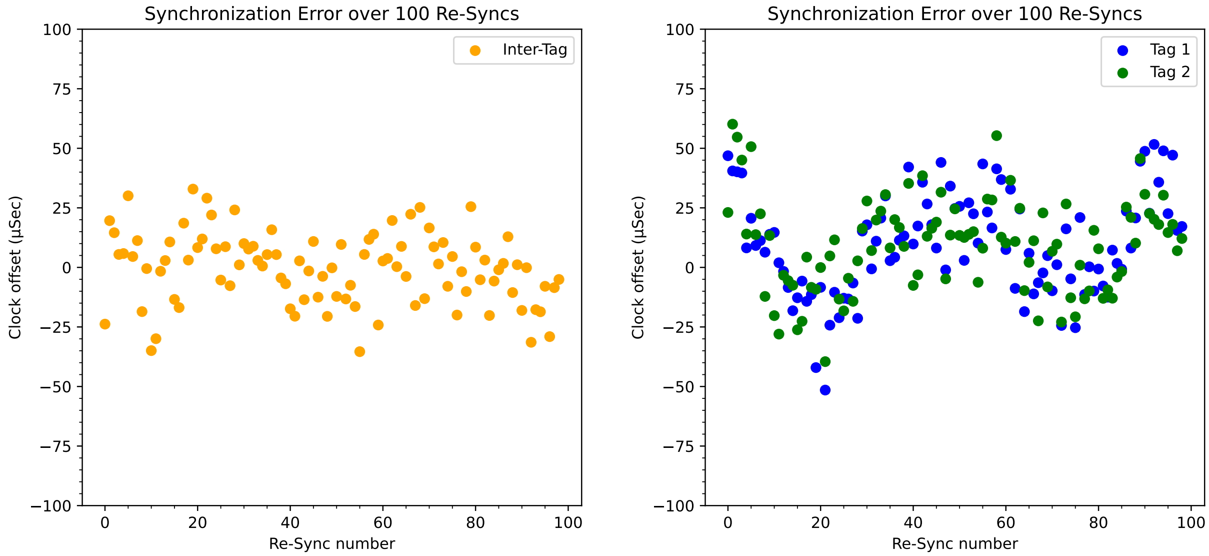

Figure 4.2: Synchronization Error over 100 Re-syncs [Far-Field]

In addition to these plots, we have also created histograms to visualize the same data sets more effectively. These histograms, presented in Fig. 4.3 for the near-field scenario and Fig. 4.4 for the far-field case, further help to illustrate the distribution of synchronization errors, thus providing a clearer picture of the system performance in different operational contexts.

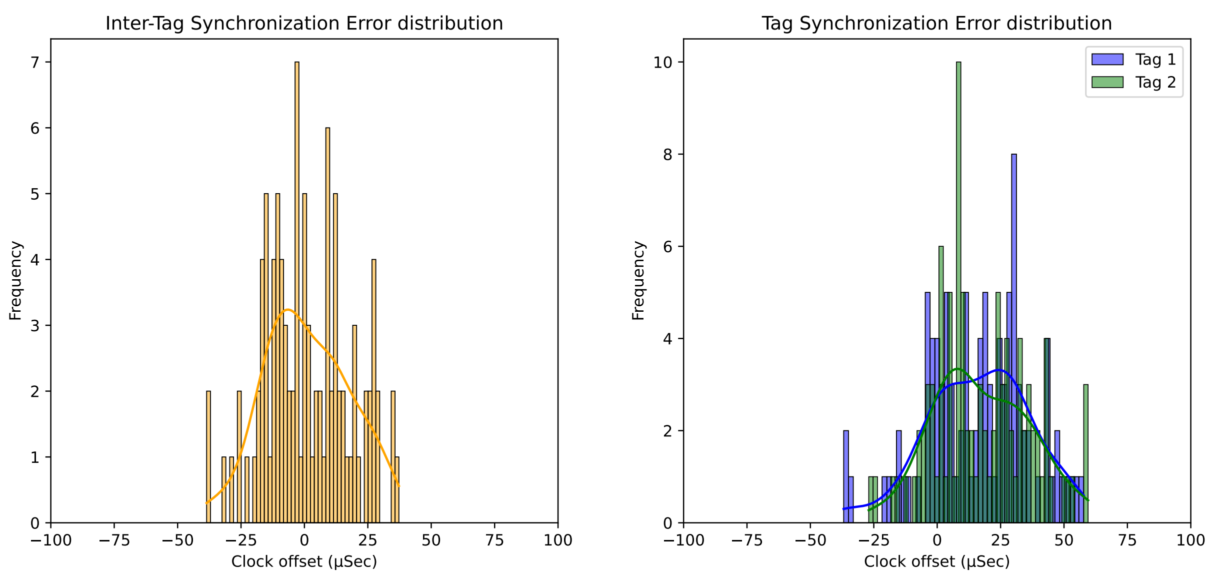

Figure 4.3: Synchronization Error Distribution [Near-Field]

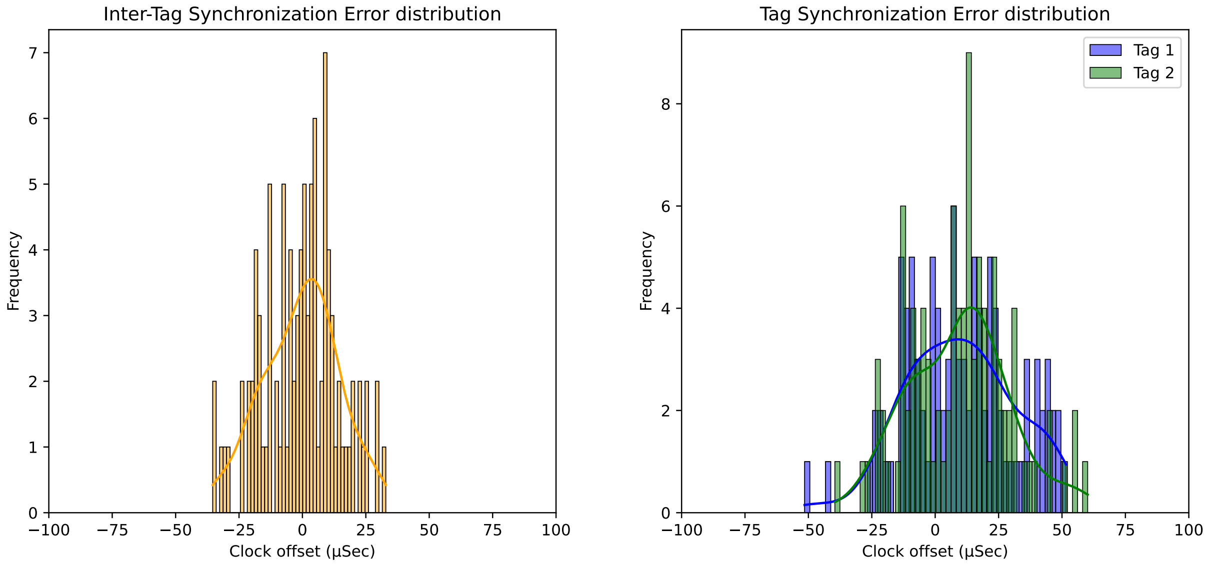

Figure 4.4: Synchronization Error Distribution [Far-Field]

4.2.4 Evaluation of Ongoing Performance

For the evaluation on the ongoing performance the Re-Sync period was set to 100 seconds to observe the long-term stability and reliability of the synchronization process over an extended period. This allowed for a comprehensive analysis of how the Re-Sync period affects the granularity of timekeeping. Different applications will require a careful consideration of the appropriate Re-Sync period based on their specific timing requirements and tolerance for synchronization errors [12].

To further evaluate it, the clocks on the nodes as well as the coordinator node were sampled using a 16Mhz logic analyzer. Table 4.3 shows the clock skew rate for Tag 1 & 2 over 3 trials of 100s for the near-field case and Table 4.3 for the far-field case. The results indicate a minor variation in clock skew, highlighting the impact of imperfection of crystal oscillators in peripheral nodes. This variation can lead to significant discrepancies in time-sensitive applications, necessitating the implementation of corrective measures such as clock drift compensation algorithms.

The graph in Fig. 4.5 displays how the clock offset between the coordinator node and

Table 4.3: Near Field node skew rate (μs/s) over 3 trials

Trial	Tag 1	Tag 2	Inter-Tag
1	14.10	15.49	1.39
2	14.13	15.59	1.46
3	14.20	15.76	1.55

Table 4.4: Far Field node skew rate (μs/s) over 3 trials

Trial	Tag 1	Tag 2	Inter-Tag
1	13.98	15.60	1.59
2	14.03	15.66	1.62
3	14.00	15.49	1.49

Node 1 and Node 2 and Inter-node clock offset evolve over time for the near-field case. The same is analysis for the far-field case is done in Fig. 4.6. A first degree polynomial is used to fit the points to give us an estimate of the skew rate. The skew rate is found to be consistent across both scenarios, suggesting that while the absolute values might differ, the underlying behavior of clock drift remains similar.

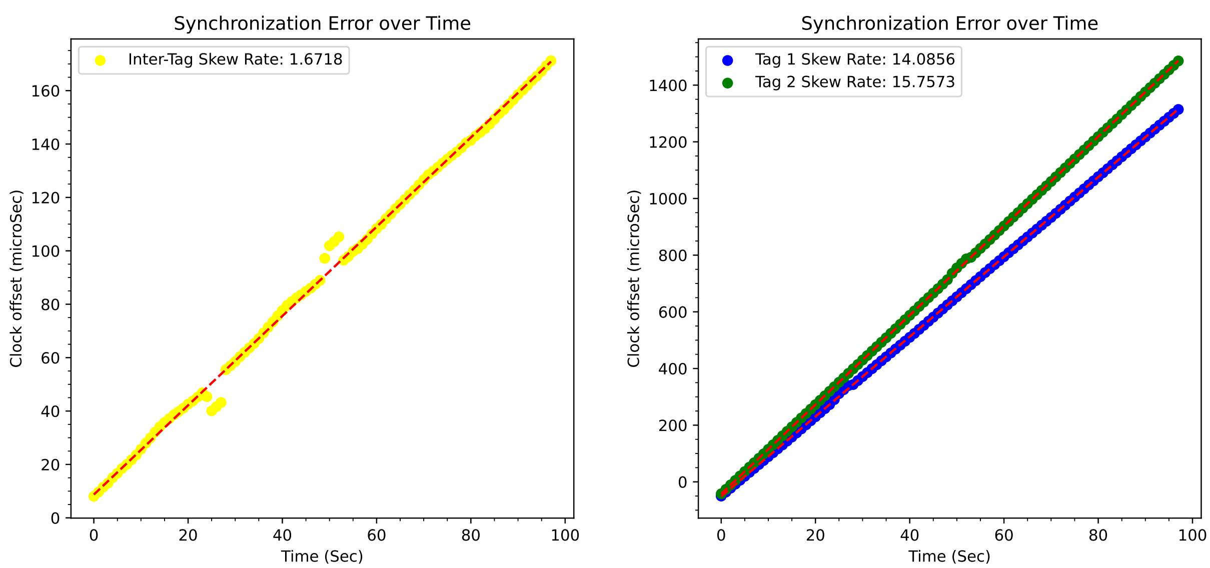

Figure 4.5: Skew Rate [Near Field]

The graph in Fig. 4.5 revealed that Node 1 exhibited a skew rate of 14.04 microseconds/s from the coordinator, while Node 2 displayed a slightly higher deviation of 15.66 microseconds/s and an inter-node skew rate of 1.62 microsec/sec ,very low but deceptive as it is a consequence of both the clocks racing at a similar speed. That would not be the case if one was lagging and the other racing compared to the coordinator clock. And therefore, the re-synchronization period need to be set according to the time-sync accuracy requirement of an application. If an applications requires nodes to be slotted at 1ms distance we need to have an accuracy in timekeeping of half a slot width which is 0.5ms. That means with a 15us/s drift we need to re-synchronize every 30s. But if an application requires 10ms slot widths we can re-synchronize every 5 minutes.

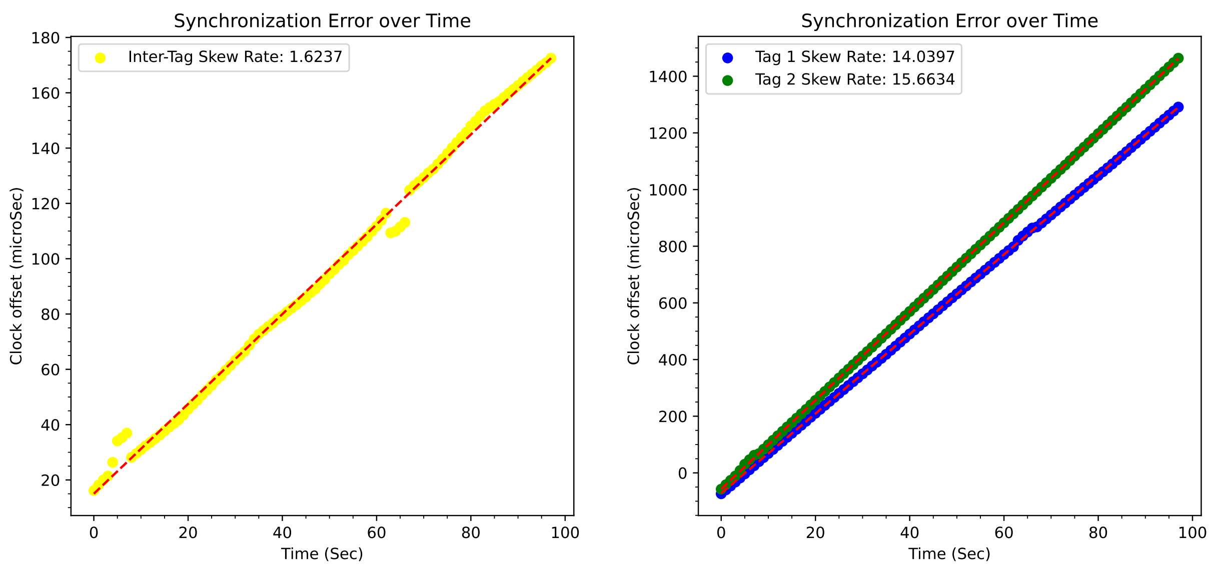

Figure 4.6: Skew Rate [Far Field]

4.3 Scheduling and MAC

With the evaluation of the synchronization accuracy, it is now possible to precisely characterize the performance of the MAC protocol and the scheduling mechanisms that are in place. This evaluation allows for a detailed analysis of how well the MAC protocol functions in terms of managing resources and coordinating communication among nodes within the network. Numerous metrics are available that can be utilized to evaluate the performance of the MAC protocol, providing insights into its effectiveness and efficiency in various aspects of operation. These metrics serve as valuable tools for assessing the overall performance of the MAC protocol and identifying areas for improvement or optimization. The main metrics that we focus on are:

1. Channel Utilization: which evaluates how efficiently the available bandwidth is utilized by the MAC protocol. A high-quality MAC protocol should aim to maximize the number of successful transmissions within a given bandwidth, thereby accommodating as many users as possible and optimizing network performance.

2. Packet Delivery Ratio (PDR): is a metric that provides valuable insights into the reliability of the communication link established by the MAC protocol. It represents the ratio of successfully received data packets to the total number of packets transmitted, indicating how well the MAC protocol handles packet loss and ensures efficient data delivery.

3. Scalability: is another crucial aspect to consider when evaluating the performance of a MAC protocol, especially as networks expand in size and node density increases. A robust MAC protocol should be able to adapt to changes in network topology and traffic load while maintaining high levels of performance. This adaptability is essential for ensuring energy efficiency, throughput, and latency remain stable even as the network grows and evolves over time.

4.3.1 UWB RTLS Case Study

To evaluate the effectiveness of the MAC protocol, the system was utilized to develop a MAC for a UWB based localization system [44] that is based on a side-channel approach. UWB technology functions with a considerably wider bandwidth in comparison to traditional narrowband systems, with the Federal Communications Commission (FCC) permitting unlicensed UWB operation within the 3.1-10.6 GHz frequency band. A well-designed MAC protocol plays a critical role in efficiently managing this extensive spectrum, enabling multiple users to utilize the bandwidth simultaneously without causing excessive levels of interference [13]. In comparison, LoRa and UWB operate at frequencies of 900 MHz and 3.5 GHz respectively, enabling them to coexist with minimal levels of interference. Details regarding their individual characteristics are mentioned in Table. 4.5. The MAC protocol based on side-channel communication was intricately crafted to facilitate effective communication among devices within the UWB localization system, thus optimizing bandwidth utilization and minimizing potential interference issues.

Table 4.5: Comparison of UWB and LoRa characteristics.

	UWB	**LoRa**
Center Freq.	3.4 GHz	930 MHz
Bandwidth	499.2 MHz	125 kHz
Preamble Len.	1024	14
Spreading Factor	-	7

Setup:

The LoRa coordinator serves as the central controller responsible for the initialization, discovery, and onboarding of all the tags within the given environment. The UWB modules, on the other hand, have been leveraged using the EVB1000 model and have been meticulously configured with the specific parameters outlined in Table 4.5. Each tag in this system has been setup to transmit 'blink' packets, with each frame it sends out carrying a payload of 14 bytes. In parallel, the LoRa tags have tasked with capturing scheduling information and time-sync packets originating from the coordinator module. This functionality enables them to effectively manage the UWB transmit slots and provide crucial support in terms of medium access control within the system. The precise timing for initiating a UWB 'blink' transmission at the designated time slot is triggered by the raising of an interrupt pin by LoRa RX.

Results:

In order to evaluate the effectiveness of the system, a total of 10 tags were set up to transmit data at a frequency of 100 Hz for a duration of thirty minutes. The analysis presented in Fig. 4.7 showcases the success rate of packet transmissions, indicating that over 99.5% of the packets were successfully received by XRLoc's localization component. In contrast, without a MAC protocol in place, the average success rate dropped to 76%, with fluctuations ranging from 56% to 87%.

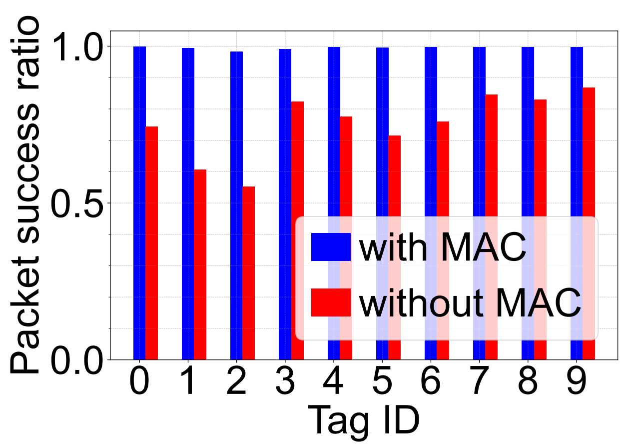

Figure 4.7: Packet success rate with MAC protocol (blue) and without it (red) tested across ten tags

A closer examination of the performance of individual tags revealed interesting insights. Fig. 4.8 displays the packet arrival rates over the thirty-minute period for both the best and worst-performing tags. It was observed that Tag 09 experienced prolonged periods of packet delivery failures, potentially due to collisions with transmissions from Tag 02 or other nearby tags. On the other hand, the implementation of a MAC protocol led to a consistent flow of packet arrivals.

These findings underscore the importance of employing a MAC protocol to ensure the smooth tracking and localization of multiple tags at high transmission rates. The data clearly demonstrates that the utilization of a MAC protocol is essential for optimizing the efficiency of the system in handling multiple tag transmissions.

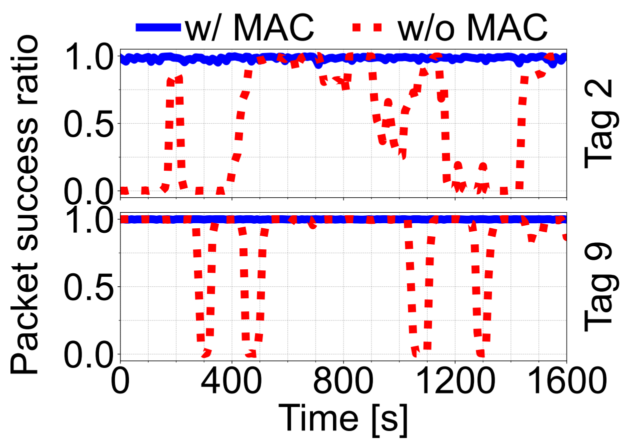

Figure 4.8: packet success ratio over time for Tag 2 (best performing without MAC) and Tag 9 (worst performing without MAC). In all cases, MAC protocol provides a success rate of over 99.5%.

Chapter 4, in part, is a reprint of the material as it appears in ”XRLoc: Accurate UWB Localization to Realize XR Deployments” by Aditya Arun, Shunsuke Saruwatari, Sureel Shah, which was published at Conference on Embedded Network Sensor Systems (SENSYS), 2023. This thesis author was the co-primary investigator and the author of this paper.

4.4 Energy Efficiency

Energy efficiency is one of the most crucial parameter when it comes to WSNs. WSNs are designed to operate with minimal power consumption in order to prolong the network's lifespan and reduce the need for frequent battery replacements. Thus, the characterization of the framework's energy model is essential.

4.4.1 Energy Model

To calculate the duty cycle and lifetime of the node, the power consumption of all different states is evaluated at a supply voltage of 3.3V. The analysis pertains to only the peripheral nodes, as the coordinator node and concentrator nodes are assumed to be connected to a stable power source, requiring a different set of considerations for energy optimization within the WSN framework. Semtex, the manufacturer of SX1272 modules have provided a Modem Energy Calculator tool [15] that allows estimation of the power consumption of transmit or a receive based on various parameters such as payload size, transmission power levels, Spreading Factor, Preamble Length, Bandwidth, Center Frequency and coding rate. Table 4.6 shows the timing and power consumption for different operating modes of the system, Table 3.2 can be referred to for the characteristics of the packets. For all the RX operations, the CAD timing and current computation is already accounted for.

To more realistically create the lifecycle of a peripheral node, we shall use a periodic sensing application where data from a TMP117 [19] temperature sensor is sampled and transmitted.

The energy consumption of the peripheral nodes E_{Total} can be expressed as :

$$E_{Total} = E_{Active} + E_{Sleep} + E_{Bat\_Discharge} \tag{4.1}$$

E_{Active} represents the energy consumed when the system is active, while E_{Sleep} accounts for the energy used when the node is in a low-power state, significantly reducing overall energy expenditure.

Energy in the sleep mode can be further decomposed as:

$$E_{Sleep} = E_{Radio\_Sleep} + E_{CPU\_Sleep} + E_{Sensor\_Sleep} \tag{4.2}$$

where E_{Radio_Sleep} denotes the energy consumed by the radio module during sleep mode, E_{CPU_Sleep} refers to the energy used by the central processing unit in its low-power state, and E_{Sensor_Sleep} indicates the energy drawn by the sensors when they are inactive.

Energy in the active mode can be further decomposed as:

$$E_{Active} = E_{Sample} + E_{rx} + E_{tx} + E_{listen} + E_{CPU\_Active} \tag{4.3}$$

where E_{Sample} represents the energy consumed during data sampling, E_{rx} is the energy used for receiving data, E_{tx} accounts for the energy spent on transmitting data, E_{listen} denotes the energy utilized while listening for incoming signals, and E_{CPU_Active} reflects the power drawn by the CPU when it is actively processing tasks.

To determine the lifetime of the system (LT), the battery capacity is E_{Bat} is divided by the energy consumption per day E_{Day} as follows:

$$LT = \frac{E_{Bat}}{E_{Day}} \tag{4.4}$$

This calculation provides insight into the operational efficiency and sustainability of the network, allowing for adjustments in power management strategies as needed.

4.4.2 Analysis of Distributed Sensing Scenario

In this section, we will model and evaluate a distributed sensing use-case where a temperature sensor is sampled and data is transmitted to comprehend how different parameters affect

Table 4.6: Operation Mode Timings and Current Consumption

Operation	Time	Symbol (Time)	I	Symbol (Current)
Radio Sleep [16]	-	T_{R_Sleep}	0.1 µA	I_{R_Sleep}
Radio Listen [16]	20 ms	T_{R_Listen}	1.5 µA	I_{R_Listen}
Radio CAD [16]	1.8 ms	T_{CAD}	9.28 mA	I_{CAD}
SYNC Packet RX [15]	20.5 ms	T_{RX_Sync}	20 mA	I_{RX_Sync}
Announcement Packet TX [15]	18.7 ms	T_{TX_Ann}	35 mA	I_{TX_Ann}
Invitation Packet RX [15]	41 ms	T_{RX_INN}	20 mA	I_{RX_Inv}
8 Byte Packet TX [15]	34 ms	T_{TX_Data}	35 mA	I_{TX_Data}
Sample Sensors [19]	15.5 ms	T_{Sensor_sample}	135 µA	I_{Sensor_sample}
Sensor Sleep [19]	-	T_{Sensor_Sleep}	1.25 µA	I_{Sensor_Sleep}
CPU Sleep [18]	-	T_{CPU_Sleep}	0.86 µA	I_{CPU_Sleep}
CPU Active [18]	-	T_{CPU_Active}	2.97 mA	I_{CPU_Active}
Battery Self Discharge [43]	-	$T_{Bat\_Discharge}$	7.5 µA	$I_{Bat\_Discharge}$

system performance and energy use. By modifying elements such as transmission intervals, re-synchronization intervals, and sampling rates, we will determine the best configurations that balance efficiency with responsiveness. In this scenario, nodes progress from the Discovery phase to the On-Boarded state. For each scenario, the energy consumption in the onboarded state will differ depending on the data transmission frequency and the complexity of the tasks being executed.

Here, the peripheral node is connected to a TMP117 Temperature sensor and a 950mAh battery as shown in Fig. 4.9, receiving scheduling and time-sync from the coordinator, and transmitting data to a LoRa gateway. The system is evaluated for different Re-Synchronization and sampling periods, with each sensor sampling followed by a 4-byte packet transmission by the LoRa Module. After initialization, unscheduled nodes enter the 'discovery state', listening for a SYNC packet from the coordinator. Upon receiving the SYNC packet, nodes transmit an

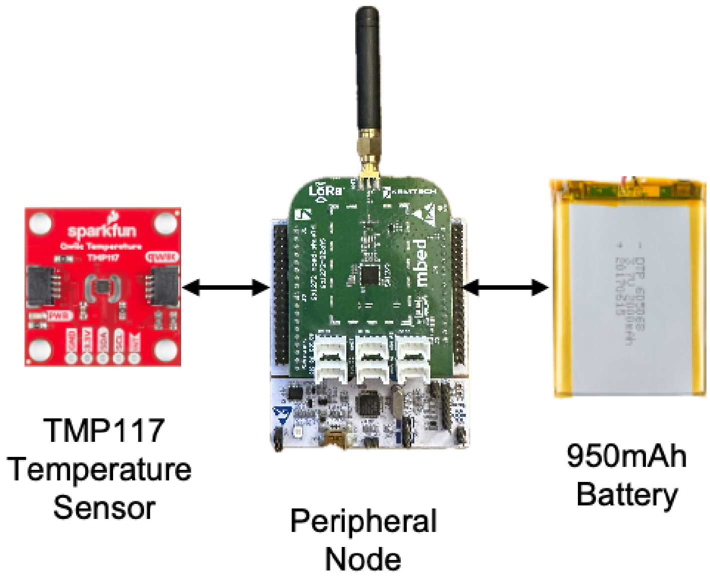

Figure 4.9: Distributed Sensing Tag

announcement packet to the Coordinator, which responds with an invitation packet containing slotting information, thus elevating the nodes to the 'onboard state', a process done once per node's lifecycle. The equation for it is given by:

$$E_{Discovery} = E_{RX\_Sync} + E_{TX\_Ann} + E_{RX\_Inv} + E_{listen} + E_{CPU\_Active} \tag{4.5}$$

For a 1 second superframe this turns out to be: 3.52J with the CPU in active phase for entire superframe.

4.4.3 Scenario 1 Energy model:

The energy consumption of the system is modeled based on the power drawn during various operational states, which include CPU activity, sensor sampling, radio listening, and sleep modes. The total energy consumed per day is calculated as the sum of the energy consumed during resynchronization, sensor sampling, packet transmission, and system sleep periods. The system is initialized with a fixed energy cost, and resynchronization and sampling intervals are variable parameters.

Discovery Energy

The system consumes a fixed amount of energy upon initialization:

$$E_{Discovery} = 3.52J \tag{4.6}$$

Resynchronization Energy

Resynchronization occurs at a specified interval T_{resync} in minutes. Each resynchronization consists of radio listening and SYNC packet reception. The energy consumed during a single resynchronization event is:

$$E_{resync} = (I_{R\_listen} \times T_{R\_listen} + I_{RX\_Sync} \times T_{RX\_Sync}) \times V \tag{4.7}$$

' Additionally, the CPU is active during the resynchronization event, consuming:

$$E_{CPU\_resync} = I_{CPU\_Active} \times V \times (T_{R\_listen} + T_{RX\_Sync}) \tag{4.8}$$

The total daily energy consumption for resynchronization is:

$$E_{Resync\_daily} = N_{resync} \times (E_{resync} + E_{CPU\_Resync}) \tag{4.9}$$

Where $N_{resync} = \frac{24 \times 60}{T_{resync}}$ is the number of resynchronizations per day.

Sensor Sampling and Transmission Energy

Sensor sampling and 8-byte packet transmission occur at a customizable interval T_{sample}. The energy consumed during each event is:

$$E_{Sample\_tx} = (I_{Sensor\_sample} \times T_{Sensor\_sample} + I_{TX\_data} \times T_{TX\_data}) \times V \tag{4.10}$$

The CPU is also active during this period, consuming:

$$E_{CPU\_sample\_tx} = I_{CPU\_Active} \times V \times (T_{Sensor\_sample} + T_{TX\_data}) \tag{4.11}$$

The total daily energy consumption for sampling and transmission is:

$$E_{Sample\_tx\_daily} = N_{sample} \times (E_{Sample\_tx} + E_{CPU\_Sample\_tx}) \tag{4.12}$$

Where $N_{sample} = \frac{24\times60}{T_{sample}}$ is the number of sampling events per day.

Sleep Energy

When the CPU, sensor, and radio are inactive, they consume power in sleep mode. The sleep time per day is the total time available in a day minus the time spent on active operations:

$$T_{sleep\_daily} = 24 \times 60 \times 60 - T_{active\_daily} \tag{4.13}$$

The total energy consumed during sleep is:

$$E_{sleep\_daily} = (I_{CPU\_sleep} + I_{sensor\_sleep} + I_{radio\_sleep}) \times V \times T_{sleep\_daily} \tag{4.14}$$

Battery Self Discharge Energy

The total energy consumed due to battery self-discharge is:

$$E_{Bat\_Discharge} = I_{Bat\_Discharge} \times (T_{sleep\_daily}) \tag{4.15}$$

Total Daily Energy Consumption

The total energy consumption for the system per day is the sum of the initialization energy, resynchronization energy, sampling and transmission energy, and sleep energy:

$$E_{total\_daily} = E_{Discovery} + E_{resync\_daily} + E_{sample\_tx\_daily} + E_{sleep\_daily} + E_{Bat\_Discharge} \tag{4.16}$$

Battery Life Estimation

Given a battery capacity $C_{battery}$ (in Ah) and voltage V, the total energy stored in the battery is:

$$E_{battery} = C_{battery} \times V \times 3600 \tag{4.17}$$

The battery life, in days, can be estimated by Equation. 4.4

With a 950 mAh in the peripheral nodes, the battery life expectations versus the sampling period are reflected in the graph in Fig. 4.10. The sampling period is plotted on the x-axis with the corresponding expected battery life on the y-axis. We can see that even with a sampling period of 60s and a re-synchronization period of 5 minutes, the node life turns out to be 4 years, which is significant. This graph is generated by using Low Power Listening (LPL) and duty cycling as part of the model. However, without these strategies, the results would change drastically. With the CPU being active for the entire lifecycle, the battery would last just 13 days for a sampling period of 60 seconds and a resynchronization period of 5 minutes.

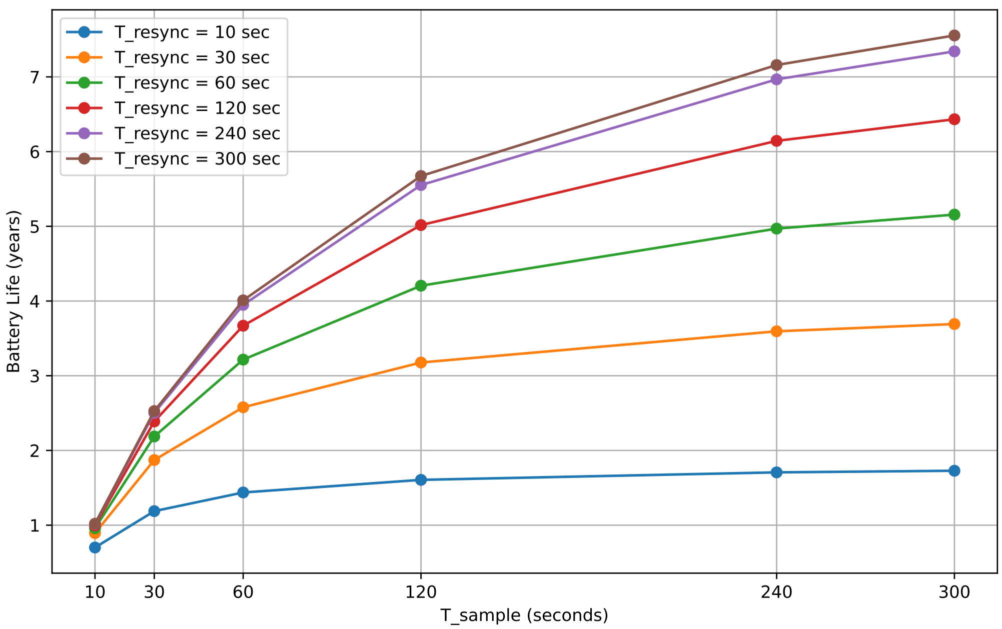

Figure 4.10: Battery life (years) Vs T_Sample

Chapter 5

Conclusion

5.1 Summary of findings

As we have delved into throughout this work, the challenges stemming from a lack of synchronization in Wireless Sensor Networks (WSNs) extend beyond mere technical hurdles; they represent fundamental obstacles to both scalability and operational efficiency. It has become increasingly clear that desynchronized clocks can result in significant data misalignment, complicating the accurate interpretation of intricate events. This misalignment not only hinders real-time data analysis but also undermines the reliability of the information being gathered. Furthermore, the absence of effective synchronization transforms scheduling into a precarious endeavor, often leading to inefficient energy use and a heightened risk of data collisions, which can severely compromise network performance. In response to these critical issues, our proposed solution innovatively combines time synchronization and scheduling mechanisms. This integration is pivotal in enhancing the scalability and operational efficiency of wireless sensor networks. By addressing these core challenges, our approach not only meets the immediate demands of current network configurations but also positions itself for future growth and expansion. This adaptability is essential for fostering sustainable technological advancement within WSN applications, ensuring that they can evolve alongside increasing demands and complexities. Moreover, the incorporation of LoRa technology, alongside the utilization of consumer prototyping boards, has allowed us to optimize our framework for both performance and ease of deployment. This strategic choice ensures that our system can be swiftly adapted to satisfy a diverse array of application requirements, enhancing its practicality and user-friendliness. Key findings from our research include:

1. **Enhanced Time Synchronization:** The introduction of a novel Medium Access Control (MAC) protocol has led to a remarkable enhancement in time synchronization across the network, achieving an impressive sub-microsecond accuracy. This precision is crucial for applications requiring high levels of temporal coordination.

2. **Improved Energy Efficiency:** The implementation of low-power listening and duty cycling strategies has significantly boosted the energy efficiency of WSNs. These strategies are instrumental in prolonging the operational lifespan of network nodes, thereby reducing maintenance costs and improving overall system longevity.

3. **Versatile Integration Capabilities:** The versatility of the MAC protocol has facilitated seamless integration with various systems, demonstrating its adaptability across a wide spectrum of Internet of Things (IoT) applications. This flexibility is vital for ensuring that our solution can cater to the diverse needs of different industries and technological environments. The integration of side-channeling also allows for secondary communication paths within the network, enhancing data transmission reliability and enabling more robust network performance even in challenging conditions.

Overall, this comprehensive approach not only addresses the immediate synchronization challenges faced by WSNs but also lays the groundwork for a robust, scalable, and efficient network infrastructure that can support future innovations and applications in the field.

5.2 Novel Contribution

This thesis introduces several key innovations to Wireless Sensor Networks (WSNs) that significantly advance both synchronization accuracy and network resource optimization while ensuring energy efficiency. The contributions of this work are as follows:

1. **One-Shot Synchronization Mechanism:** A one-shot synchronization method was developed, which aligns sensor nodes within sub-microsecond discrepancies. This mechanism achieves global synchronization across thousands of nodes with an accuracy of ±20 microseconds. By broadcasting synchronization packets through a single one-way message transfer, the framework ensures that all nodes remain precisely aligned, a crucial factor for applications requiring coordinated operations. This high precision makes the solution suitable for large-scale WSN deployments with minimal synchronization overhead.

2. **TDMA-Based Scheduling Protocol:** A robust scheduling protocol based on Time Division Multiple Access (TDMA) was designed to optimize network resource allocation and minimize collisions. The modular framework leverages this TDMA protocol to efficiently slot nodes into a 1-second superframe, accommodating tens to thousands of nodes simultaneously. The precision in timing and the effective allocation of time slots ensure predictable throughput and prevent communication interference, even in dense network environments. This approach also allows for seamless scalability, supporting networks ranging from small-scale setups to extensive deployments with thousands of nodes.

3. **Experimental Validation and Benchmarking:** A detailed experimental evaluation was conducted to validate the framework's performance across several key metrics, including synchronization accuracy, throughput, and energy efficiency. The experiments demonstrated the effectiveness of the one-shot synchronization and TDMA-based scheduling protocol in maintaining network stability and reliability. Additionally, the results serve as a benchmark

for future enhancements, providing a solid foundation for further research in scalable and energy-efficient WSN designs.

4. **Low-Power Listening and Duty Cycling:** To improve energy efficiency, the framework implements Low-Power Listening (LPL) and microcontroller unit (MCU) duty cycling. These features enable nodes to remain in a low-power sleep mode when not actively communicating, significantly reducing overall energy consumption. This approach extends the battery life of the sensor nodes, making the framework highly suitable for long-term, remote deployments where power resources are limited.

These contributions together provide a comprehensive and scalable solution for synchronized, low-power, and collision-free communication in large-scale WSNs, establishing a new standard for precision and efficiency in network design.

5.2.1 Potential Applications for the Framework

The modular framework developed in this research offers a robust and scalable solution for Wireless Sensor Networks (WSNs), making it highly adaptable for a variety of applications across different industries. Its key features, such as precise time synchronization, low power consumption, and flexible scheduling, make it particularly suitable for environments where reliability, energy efficiency, and scalability are critical. Below are several applications where this framework can provide significant value:

1. **Wildlife Tracking and Behavior Monitoring**: The framework can be used in battery-powered collars equipped with LoRa and Ultra-Wideband (UWB) technologies to track the real-time location and monitor the behavior of wildlife over expansive areas. This application is particularly useful for conservation efforts, where tracking endangered species and understanding their movement patterns is essential for preserving habitats and ensuring species survival. The low power consumption of the framework ensures that the collars

can operate for extended periods without frequent battery replacements, making long-term monitoring feasible in remote locations.

2. **Precision Agriculture**: In agricultural settings, the framework supports the deployment of battery-powered sensors across farmlands to gather critical data on soil moisture, temperature, humidity, and other environmental parameters. These sensors provide real-time insights into crop health and soil conditions, enabling farmers to optimize irrigation, fertilization, and other farming practices. By leveraging the low-power features of the framework, the sensors can operate autonomously for long periods, minimizing maintenance costs and improving the efficiency of farm management.

3. **Wildfire Detection in Remote Forest Areas**: Forest management agencies can install battery-powered sensors in remote forest areas to monitor environmental conditions such as temperature, humidity, and smoke levels. The framework allows for the early detection of changes that may indicate the onset of wildfires, providing critical early warnings to firefighting teams. By optimizing battery usage through low-power listening and duty cycling, the sensors can remain operational for extended periods, even in harsh environmental conditions where access to power sources is limited.

4. **Smart Waste Management**: Battery-powered sensors can be deployed in waste containers to monitor fill levels and send real-time data to waste management systems. By utilizing the framework's energy-efficient communication protocol, these sensors can transmit data over long distances to central hubs, helping optimize waste collection routes and schedules. This reduces unnecessary trips, saves fuel, and improves the overall efficiency of waste collection operations, while also contributing to a reduction in carbon emissions.

5. **Asset Tracking in Industrial Facilities**: Large industrial facilities can utilize battery-powered trackers based on the modular framework to monitor the location and usage of tools, equipment, and other assets. This helps prevent loss, ensure the optimal usage of

equipment, and reduce downtime due to misplaced tools. The scalability and low energy consumption of the framework enable facilities to manage a large number of assets over extensive areas, providing real-time visibility into their operations.

6. **Smart Parking Management**: The framework can be applied to battery-operated sensors installed in parking spaces to detect vehicle occupancy. These sensors can communicate real-time data to drivers, guiding them to available parking spots, thus reducing congestion and emissions in urban areas. The energy-efficient features of the framework allow the sensors to remain active for long periods, providing reliable parking management without frequent battery replacements.

7. **Remote Health Monitoring**: In healthcare, the framework supports the deployment of battery-powered health monitoring devices that can continuously monitor patients' vital signs and communicate data back to healthcare providers. These devices enable real-time health monitoring without the need for frequent charging, improving patient comfort and reducing the burden on healthcare systems. The low power consumption and reliable communication protocols make this framework particularly suitable for wearable medical devices, ensuring they remain operational for extended periods between recharges.

The applications outlined above demonstrate the versatility and broad utility of the modular framework developed in this thesis. From wildlife conservation to healthcare, smart cities and industrial asset management, the ability of the framework to provide reliable, low-power, and scalable solutions makes it an ideal choice for a wide range of WSN deployments. As IoT applications continue to expand, this framework offers a powerful tool to address the unique challenges posed by diverse industries, improve efficiency, and enable smarter data-driven decision making.

5.3 Future Works

While the modular framework developed in this thesis has demonstrated significant improvements in time synchronization, energy efficiency, and network scalability, there are several potential areas for future work. Expanding on the applications and novel contributions of this framework, further research and development could explore the following directions to enhance its capabilities and extend its applicability:

1. **Dynamic Slot Scheduling and Adaptation**: While the current TDMA-based scheduling protocol efficiently manages time slots and minimizes collisions, there is potential to develop more advanced dynamic slot scheduling algorithms. These algorithms could adapt to real-time network conditions, such as varying data traffic, node energy levels, or environmental factors, to further optimize network performance. This would be particularly beneficial in scenarios with fluctuating data transmission needs, such as precision agriculture or smart parking, where certain nodes may require more frequent data transmission at different times.

2. **Enhanced Multi-Hop Communication**: The current framework operates effectively in a star topology, but future work could investigate the implementation of multi-hop communication to extend the network's range further, especially in large, expansive areas like farmlands or industrial sites. This would involve developing routing protocols that maintain the framework's synchronization accuracy and low energy consumption while enabling data to hop between nodes before reaching the Gateway.

3. **Improving Synchronization for High-Mobility Applications**: As part of the framework's future development, enhancing the synchronization accuracy and robustness in highly mobile applications could be explored. For example, wildlife tracking collars and remote health monitoring devices for patients on the move may require additional synchronization improvements to maintain sub-microsecond alignment as nodes move in and out of

communication range. Techniques such as rapid re-synchronization upon re-entry into the network or more sophisticated prediction-based skew compensation could be investigated.

4. **Support for Advanced Sensing Capabilities**: The framework could be extended to support advanced sensing and data processing capabilities at the edge, allowing nodes to perform preliminary analysis of sensor data before transmission. This would be particularly useful in applications such as wildfire detection or industrial asset tracking, where local processing of data could enable faster response times or reduce the amount of data that needs to be transmitted back to the Gateway, conserving energy and bandwidth.

5. **Testing in Diverse Real-World Environments**: While the framework has been validated in controlled experimental settings, future work should focus on deploying the system in more diverse and challenging real-world environments. For example, deploying the system in dense urban areas, remote forests, or extreme industrial environments could provide valuable insights into its performance under different physical conditions. Field tests would also help refine the system's energy consumption, reliability, and scalability in larger deployments.

The novel contributions of this thesis have provided a solid foundation for time synchronization, energy efficiency, and scalability in Wireless Sensor Networks, making it highly applicable to a variety of real-world scenarios. Future research and development can build on this foundation by exploring more advanced energy-saving techniques, improved dynamic scheduling, multi-hop communication, and support for advanced sensing capabilities to name a few. These improvements will not only increase the framework's robustness but also expand its potential to address a wider range of IoT and WSN applications, from large-scale environmental monitoring to complex industrial operations.

Bibliography

[1] H. Rajab, H. Al-Amaireh, T. Bouguera, and T. Cinkler, "Evaluation of energy consumption of LPWAN technologies," *EURASIP Journal on Wireless Communications and Networking*, vol. 2023, p. 118, Dec. 2023.

[2] M. Iqbal, A. Y. M. Abdullah, and F. Shabnam, "An application based comparative study of lpwan technologies for iot environment," in *2020 IEEE Region 10 Symposium (TENSYMP)*, pp. 1857–1860, 2020.

[3] R. D. Martin, *Human Brain Evolution in an Ecological Context*. American Museum of Natural History, 1983.

[4] J. K. Rilling and T. R. Insel, "The primate neocortex in comparative perspective using magnetic resonance imaging," *Journal of Human Evolution*, vol. 37, no. 2, pp. 191–223, 1999.

[5] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy, "The platforms enabling wireless sensor networks," *Commun. ACM*, vol. 47, p. 41–46, jun 2004.

[6] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[7] D. D. Geetha and N. Tabassum, "A survey on clock synchronization protocols in wireless sensor networks," in *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, pp. 504–509, 2017.

[8] I.-K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y.-C. Wu, "Clock synchronization in wireless sensor networks: An overview," *Sensors*, vol. 9, no. 1, pp. 56–85, 2009.

[9] "SX1272MB2xAS / SX1272MB2DAS | Mbed."

[10] "SX1272 | Long Range, Low Power RF Transceiver 860-1000MHz | Semtech."

[11] "NUCLEO-L073RZ | Mbed."

[12] M. Rizzi, P. Ferrari, A. Flammini, and E. Sisinni, "Evaluation of the IoT LoRaWAN Solution for Distributed Measurement Applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, pp. 3340–3349, Dec. 2017.

[13] N. Macoir, M. Ridolfi, J. Rossey, I. Moerman, and E. D. Poorter, "MAC Protocol for Supporting Multiple Roaming Users in Multi-Cell UWB Localization Networks,"

[14] "LoRa PHY | Semtech."

[15] Semtech, *LoRa Modem Design Guide*, July 2013. 1–9.

[16] Semtech, *SX1272/73 Datasheet*, March 2017.

[17] Semtech, *LoRa and LoRaWan: A Technical Overview*, December 2019.

[18] STM, *STM32L073x8 STM32L073xB STM32L073xZ Datasheet*, September 2022.

[19] Texas Instruments, *TMP117 Datasheet*, June 2021.

[20] 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.

[21] X. Ouyang, O. A. Dobre, Y. L. Guan, and J. Zhao, "Chirp spread spectrum toward the nyquist signaling rate—orthogonality condition and applications," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1488–1492, 2017.

[22] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Fair adaptive data rate allocation and power control in lorawan," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 14–15, 2018.

[23] G. Kim and J. Ahn, "On-demand synchronous x-mac protocol," in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 1–6, 2016.

[24] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient collision-free medium access control for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, (New York, NY, USA), p. 181–192, Association for Computing Machinery, 2003.

[25] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, (New York, NY, USA), p. 95–107, Association for Computing Machinery, 2004.

[26] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1567–1576 vol.3, 2002.

[27] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, (New York, NY, USA), p. 171–180, Association for Computing Machinery, 2003.

[28] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, (New York, NY, USA), p. 307–320, Association for Computing Machinery, 2006.

[29] A. Gudnavar and N. Manjanaik, "A survey on energy-efficient mac protocols for wireless sensor networks," in *Smart Sensors Measurements and Instrumentation* (S. K V and K. Rao, eds.), (Singapore), pp. 177–188, Springer Singapore, 2021.

[30] V. Rajendran, J. Garcia-Luna-Aveces, and K. Obraczka, "Energy-efficient, application-aware medium access for sensor networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.*, pp. 8 pp.–630, 2005.

[31] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-mac: a hybrid mac for wireless sensor networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, (New York, NY, USA), p. 90–101, Association for Computing Machinery, 2005.

[32] D. Djenouri and M. Bagaa, "Synchronization protocols and implementation issues in wireless sensor networks: A review," *IEEE Systems Journal*, vol. 10, no. 2, pp. 617–627, 2016.

[33] I.-K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y.-C. Wu, "Clock synchronization in wireless sensor networks: An overview," *Sensors*, vol. 9, no. 1, pp. 56–85, 2009.

[34] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, (New York, NY, USA), p. 138–149, Association for Computing Machinery, 2003.

[35] M. Scanlon, C. Reiff, and L. Solomon, "Aerostat acoustic payload for transient and helicopter detection," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VI* (E. M. Carapezza, ed.), vol. 6538, p. 65380H, International Society for Optics and Photonics, SPIE, 2007.

[36] P.-C. Hii and W.-Y. Chung, "A comprehensive ubiquitous healthcare solution on an android™ mobile device," *Sensors*, vol. 11, no. 7, pp. 6799–6815, 2011.

[37] S. Mansour, N. Nasser, L. Karim, and A. Ali, "Wireless sensor network-based air quality monitoring system," in *2014 International Conference on Computing, Networking and Communications (ICNC)*, pp. 545–550, 2014.

[38] H. Zhang and S. Yang, "Research on application of wsn in cold chain logistics' warehousing and transportation," in *Advanced Graphic Communications and Media Technologies*, pp. 589–598, Springer, 2017.

[39] M. Liya and D. Arjun, "A survey of lpwan technology in agricultural field," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 313–317, 2020.

[40] J. Sanchez-Gomez, J. Gallego-Madrid, R. Sanchez-Iborra, and A. F. Skarmeta, "Performance study of lorawan for smart-city applications," in *2019 IEEE 2nd 5G World Forum (5GWF)*, pp. 58–62, 2019.

[41] G. M. Rahman, K. A. Wahid, and A. Dinh, "Iot enabled low power and wide range wsn platform for environment monitoring application," in *2020 IEEE Region 10 Symposium (TENSYMP)*, pp. 908–911, 2020.

[42] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, (New York, NY, USA), p. 39–49, Association for Computing Machinery, 2004.

[43] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, "Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN," *Sensors*, vol. 18, p. 2104, June 2018.

[44] A. Arun, S. Saruwatari, S. Shah, and D. Bharadia, "Xrloc: Accurate uwb localization to realize xr deployments," in *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, SenSys '23, (New York, NY, USA), p. 459–473, Association for Computing Machinery, 2024.