UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

An Efficient Communication and Routing Protocol for Sustainable Smart Agriculture Sensor Networks

Nicholas Wishart
22140573

Report submitted in partial fulfilment of the requirements of the module Project (E) 448 for the degree Baccalaureus in Engineering in the Department of Electrical and Electronic Engineering at Stellenbosch University.

Supervisor: Prof R. Wolhuter

November 2024

Acknowledgements

I would like to thank my supervisor Prof. Wolhuter for giving me his knowledge and guidance during this project. My parents for providing me with support throughout my studies and Caid for always lending a hand over the years.

UNIVERSITEIT·STELLENBOSCH·UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.
2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   I agree that plagiarism is a punishable offence because it constitutes theft.
3. Ek verstaan ook dat direkte vertalings plagiaat is.
   I also understand that direct translations are plagiarism.
4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism
5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / *Student number*	Handtekening / *Signature*
22140573	NDWishart
Voorletters en van / *Initials and surname*	**Datum / *Date***
ND Wishart	15/08/2024

Abstract

English

This project investigates energy-efficient routing protocol optimization in wireless sensor networks tailored for smart agriculture applications, with a specific focus on vineyard environments. Modern agriculture demands precise monitoring of environmental factors such as soil moisture, temperature, and humidity to enhance crop quality and resource efficiency. However, deploying WSNs in remote or outdoor settings like vineyards presents unique challenges in terms of power efficiency, data reliability, and adaptive routing. By using node metrics—including battery level, received signal strength, distance, and node priority this project develops an adaptive routing algorithm designed to optimize energy consumption for scalable networks.

This research contributes to the field of smart agriculture by presenting a scalable, cost-effective WSN model optimized for vineyards, capable of reducing overall energy overhead and supporting long-term, sustainable agricultural practices.

Afrikaans

Hierdie projek ondersoek energiedoeltreffende roeteprotokoloptimering in radiogebaseerde sensor netwerke vir slim landboutoepassings, met 'n spesifieke fokus op wingerdbou. Moderne landbou vereis presiese monitering van omgewingsfaktore soos grondvog, temperatuur en humiditeit om gewaskwaliteit en hulpbronbenutting te verbeter. Die ontplooiing van WSNs in afgeleë of buitelugomgewings soos wingerde bied egter unieke uitdagings in terme van kragdoeltreffendheid, databetroubaarheid en dataroetering. Met behulp van metings van batteryvlakke, seinsterkte, afstande en nodusprioriteit by die verskillend nodes, is in hierdie projek 'n aanpasbare roeteringsalgoritme ontwerp om energieverbruik te optimeer. Die metode skaleer goed vir verskillende netwerkgroottes.

Hierdie navorsing dra by tot die veld van intelligente landbou deur 'n skaleerbare, kos-effektiewe WSN-model spesifiek vir wingerde aan te bied, wat in staat is om goeie hulpbronbestuur te optimeer, bekoste te verminder en langtermyn, volhoubare landboupraktyke te ondersteun.

Contents

List of Figures

List of Tables

Nomenclature

Variables and functions

P_{tx}	Transmission power of the node.
$RSSI$	Received Signal Strength Indicator, a measure of the power level received by a node.
L_{avg}	Average latency of data packets from source to destination.
E_{node}	Energy consumed by each node during a data-gathering sequence.
H	Number of hops in the network, indicating the number of intermediate nodes in data transmission.
QoS	Quality of Service, measured by the ratio of data requests to data replies.
DREQ	Data request packet type for node discovery and neighbor table updates.
DREP	Data reply packet type, containing sensor payload and routing metrics.
R_{cost}	Routing cost metric, determined by factors like battery level and RSSI.

Acronyms and abbreviations

ADC	Analog-to-Digital Converter
BS	Base Station
CCA	Clear Channel Assesment
CRC	Cyclic Redundancy Check
CTS	Clear-To-Send
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DREQ	Data Request Packet
DREP	Data Reply Packet
DSSS	Direct Sequence Spread Spectrum
ESP-NOW	Espressif Proprietary Protocol for Peer-to-Peer Communication
GPS	Global Positioning System
GSM	Global System for Mobile Communications
IEEE	Institute of Electrical and Electronics Engineers
LEACH	Low-Energy Adaptive Clustering Hierarchy
MAC	Medium Access Layer
MCU	Microcontroller Unit
OSI	Open Systems Interconnection Reference Model
PEGASIS	Power-Efficient Gathering in Sensor Information Systems
QoS	Quality of Service
RMS	Root Mean Squared
RSSI	Received Signal Strength Indicator
RTC	Real-Time Clock
RTS	Request-To-Send
UART	Universal Asynchronous Receiver/Transmitter
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1. Background

Wireless sensor networks (WSNs) in recent years have emerged as an important component in various fields, such as environmental monitoring, industrial automation, and precision agriculture. These networks consist of sensor nodes that communicate wirelessly to collect and transmit data over a defined area, typically relaying information back to a central system for analysis. Vineyard sensor networks, specifically, play an important role in improving the efficiency and sustainability of agriculture by providing real-time monitoring of environmental factors like temperature, humidity, and soil moisture content [1]. However, despite their promise, deploying and maintaining energy-efficient and reliable WSNs remains a significant challenge.

Energy efficiency is particularly important due to the limited power supply of sensor nodes, which are often located in remote or inaccessible areas where battery replacement is impractical [2]. Additionally, the topographical and environmental characteristics of vineyards introduce challenges such as signal interference, node placement, and routing reliability, all of which can severely affect a network's performance and energy consumption.

1.2. Problem Statement

In vineyard environments, the need for continuous, reliable, and low-power data transmission creates a demand for specialised routing protocols that can extend the network's lifetime while maintaining robust data transfer. Existing commercial agricultural WSNs offer solutions but often at the cost of adaptability, scalability and power consumption [1]. Furthermore, energy consumption can often be made worse by inefficient communication protocols and topological challenges within it's vicinity [3]. This project aims to address these issues by optimising an adaptive routing protocol with a focus on energy efficiency and scalability for vineyard-based sensor networks.

1.3. Objectives

The primary objectives of this research are to:

- Design and implement an energy-efficient routing protocol tailored for vineyard WSNs.
- Evaluate the performance of the protocol in terms of energy consumption, reliability, and scalability.
- Incorporate key sensor metrics such as node battery level, RSSI, and energy cost to the gateway in the routing decisions.
- Ensure that the protocol adapts to topological changes, such as node failures or environmental changes, to maintain network robustness.

1.4. Scope

The scope of this project is limited to the design, implementation, and simulation of a wireless sensor network deployed in vineyard environments. The hardware setup includes low-power ESP32 nodes equipped with environmental sensors to measure parameters such as temperature, humidity and pressure. The study will primarily focus on optimising communication protocols for energy efficiency while considering the practical challenges of node placement and signal propagation in a vineyard.

1.5. Summary of Work

This project proposes the development of a custom routing protocol for vineyard sensor networks, emphasising energy efficiency and adaptability. The research will involve both theoretical development and practical simulation of the proposed protocol, followed by practical implementation and field testing in a vineyard environment. The project will look at existing work in energy-aware routing but will focus on optimising for the specific challenges encountered in vineyard settings. Performance metrics, including energy consumption and network reliability will be analysed to evaluate the effectiveness of the protocol.

1.6. Project Overview

This project is structured as follows: Chapter 2 provides a literature review of related work in wireless sensor networks, energy-efficient protocols, and agricultural monitoring systems. Chapter 3 introduces the system design, including the hardware and software components used in the network. In Chapter 4, the design of the system, proposed routing protocol is detailed, along with the optimisation techniques applied. Chapter 5 presents the evaluation methodology, followed by Chapter 6 which concludes with a summary of the research and potential directions for future work.

Chapter 2

Literature Review

This chapter will provide insight to field of WSN applications for Smart Agriculture, with a focus on a vineyard setting, and discuss prior research that has been conducted in the field of routing optimisation in this area. Additionally it will look at additional relevant aspects of routing in WSNs. The research carried out here will provide a good foundation of knowledge to understand the topics covered in this project.

2.1. Smart Agriculture

2.1.1. Technological Advancements

Smart Agriculture incorporates the use of technology into farming practices to increase the sustainability and efficiency of agriculture. It uses a data-driven approach to help farmers make informed decisions regarding the management of their farms. Sensors and other tools can be used to provide valuable data for analysis. [4]

The data collected can provide detailed insight into soil moisture level, temperature, humidity, and crop health. This information can allow farmers to optimise irrigation schedules, predict disease onset, or implement fertilisation strategies amongst others. By using sensors, drones, and data analytics, precision agriculture enables farmers to be more efficient and sustainable, enhancing crop quality and yield. [5]

WSNs can enhance decision-making processes by providing insights enabling important decisions to be made. For instance, early detection of soil nutrient deficiencies or the onset of disease allows for quick intervention, reducing potential losses. [6] The scalability and flexibility of WSNs make them suitable for many different agricultural settings, from small-scale farms to large operations. Moreover, advancements in wireless communication technologies have improved the reliability and energy efficiency of WSNs, making them more accessible and practical for agricultural applications. [7]

2.1.2. Challenges in Implementation

Implementing smart agricultural practices, particularly in vineyards, presents a set of challenges that can significantly affect the performance, reliability, and energy efficiency of the network.

Costs associated with hardware procurement, installation, and setup can be restrictive for small-scale farmers or those in developing regions. Ongoing expenses for maintenance, such as battery replacement and hardware repairs, add to the total cost of ownership. [6] In harsh environmental conditions where hardware failures are more frequent, these expenses can increase, potentially outweighing the benefits of the technology. Developing cost-effective solutions that reduce both initial and maintenance expenses are necessary to encourage adoption.

As a network grows in scale, maintaining efficient routing and data aggregation becomes more complex. [8] Larger networks may experience increased latency, higher packet loss rates, and greater routing overhead if not properly managed.

2.2. Related Work

2.2.1. LEACH

A low-energy adaptive clustering hierarchy or LEACH is an application-specific architecture for WSNs. [9] [10] It primarily makes use of node clustering and data aggregation to optimise power efficiency in a network. LEACH uses a hierarchical clustering approach to reduce overall system energy consumption. It seeks to prolong the network lifetime by distributing energy consumption among all sensor nodes, reducing data transmission distances through clustering, and decreasing the amount of data that needs to be sent to the base station via data aggregation techniques. By dynamically selecting cluster heads LEACH makes sure no individual node is overburdened with message transmissions.

Method

Nodes organise themselves into clusters based on a probabilistic model with each node deciding independently whether to become a cluster-head in a given round. The role of cluster-head rotates among the nodes. This rotation is randomized to prevent energy depletion of specific nodes. Cluster heads collect data from nodes within their cluster. They aggregate this data before sending it to the data endpoint.

Results

The LEACH protocol can yield energy savings compared to traditional flat routing protocols. The localised data transmission and random cluster head selection ensures that the overall lifetime of the system is prolonged. LEACH reduces the high-power long transmissions of a system ultimately reducing the energy load on the system.

LEACH assumes that all nodes can directly communicate with the base station, which may not be feasible in large-scale networks. The random selection of cluster heads may lead to sub-optimal cluster formations, affecting network efficiency.

2.2.2. PEGASIS

PEGASIS (power-efficient gathering in sensor information systems) is a chain-based protocol that offers a different approach to the clustering used in LEACH. [11] The protocol seeks to remove the overhead of dynamically forming clusters and selecting cluster heads using a chain-based approach to transmit node data to a base station (BS). This approach intends to reduce the overall power consumption of nodes and thus prolong the lifetime of the overall system.

Method

A chain is formed among all sensor nodes using a greedy algorithm. Starting from a random node, each node connects to its nearest neighbour that has not yet been visited, continuing until all nodes are connected in a line. This approach uses a solution similar to the Travelling Salesman Problem, which is computationally intensive. In each data-gathering round, nodes receive data from one neighbour, fuse it with their own sensed data, and then transmit the aggregated data to the next neighbour along the chain. One node in the chain is designated to transmit all the data to the BS. Since the BS is located at a far distance from the sensor nodes, minimising the number of transmissions to the BS conserves significant energy.

Results

PEGASIS uses local data aggregation to reduce the amount of data transmitted to the BS and minimises communication distances between nodes. This method results in energy savings and extends the operational lifetime of the sensor network. Simulation studies have shown that PEGASIS can achieve between 100% to 300% improvement over LEACH in terms of network lifetime. However, it should be noted that this is mainly due to the dynamic behaviour when nodes run out of power. [11]

The greedy algorithm used may not always produce the most efficient chain, leading to sub-optimal performance. PEGASIS can introduce significant delays. Increased latency should be noted for applications requiring real-time data processing. Additionally, PEGASIS assumes that all nodes can communicate with the BS, this will not be suitable for larger-scale networks where not all nodes are in the range of the BS.

2.3. Additional Technologies

2.3.1. Wireless Sensor Networks

This section will look at the different aspects of ad-hoc sensor networks. The OSI model and its various layers play a part in understanding how WSNs function. A central focus of this project is on the network that allows communication between the sensor nodes. The configuration of the network has a big impact on the system attributes, including power consumption and reliability.

Advancements in technologies such as batteries, low-power electronics and radio devices have made wireless sensor networks feasible at reasonable costs. WSNs consist of a number of sensor nodes that monitor environmental or physical conditions such as temperature, humidity, and soil moisture. These nodes transmit data wirelessly to a central system for analysis. In agriculture, WSNs enable real-time monitoring and management of crops, enhancing decision-making processes and promoting efficient resource utilisation. By integrating WSNs, farmers can automate data collection, reduce labour costs, and improve crop yields through precise management. [12] In order to design power efficient routing methods for these networks we must understand the challenges faced with their implementation.

2.3.2. Challenges

Environmental Factors

Vineyards often span uneven terrain, slopes, and varying elevations, which can affect signal propagation and node placement. This can lead to signal shadowing and increased path loss, impacting the reliability of wireless communications. Strategically positioning sensor nodes to reduce these issues is helpful but challenging.

Dense vegetation, such as vines and leaves introduce additional obstacles for wireless signal transmission. Foliage can cause signal attenuation, reflection, and path propagation, leading to decreased signal strength and increased error rates. [13] This interference may force nodes into transmitting data again or increasing transmission power to maintain connections, resulting in higher energy consumption and reduced network lifespan.

Extreme environmental conditions common in agricultural settings can impact the durability and performance of sensor hardware. [6] High temperatures, humidity, and precipitation can affect electronic components and battery efficiency. Moisture can damage circuitry, while temperature fluctuations can degrade battery capacity and sensor accuracy. Ensuring that hardware is robust enough to withstand these conditions is important for reliable long-term operation.

Energy Consumption Issues

Sensor nodes typically rely on batteries or energy-harvesting methods, which are constrained in capacity. A wired power source is usually impractical, especially in large-scale deployments or hard-to-reach areas within a vineyard. While solar panels and other energy-harvesting solutions can supplement power, they introduce additional complexity and cost.

Overlapping coverage areas can cause unnecessary energy use due to multiple data transmissions in the same area, while insufficient coverage results in data loss. Therefore achieving adequate coverage without too much redundancy is challenging in vineyards due to their extensive and irregularly shaped areas.

Data transmission is typically the most energy-intensive operation of sensor nodes. [14] High data rates and frequent transmissions can quickly drain node batteries, reducing network lifespan. Implementing energy-efficient communication protocols and techniques, such as optimising transmission power and using low-power wireless technologies, is essential to minimise energy consumption. [15]

Network Reliability and Maintenance

Nodes in vineyards may be difficult to access for maintenance or battery replacement due to the vastness of the area and the presence of dense vegetation and trellises. This physical inaccessibility can lead to increased operational costs and downtime if nodes fail or require servicing.

A network must handle node failures without significant performance degradation. In agricultural environments, hardware failures can occur due to environmental stress or physical damage. Implementing fault-tolerant network designs, such as mesh networking, can enhance connectivity by allowing data to be relayed through multiple nodes, effectively bypassing failed nodes and ensuring reliable communication across the vineyard.

Communication Channel

WSNs use radio communication, which introduces challenges not found in wired networks, particularly regarding transmission collisions over a shared broadcast medium. Unlike wired networks where collision detection is straightforward, wireless networks can experience interference when multiple transmissions occur simultaneously, leading to data loss and reduced network performance.

To mitigate this issue, the IEEE 802.11 standard employs Carrier Sense Multiple Access (CSMA), a protocol derived from the ALOHA system. [16] CSMA allows nodes to listen to the communication channel before transmitting data. If a node detects that the channel is busy, it waits for a random interval before attempting to transmit again. This method reduces the likelihood of collisions by minimising the chance that multiple nodes transmit at the same time.

However, CSMA has limitations in wireless environments due to the hidden node problem. For instance, consider nodes A, B, and C as can be seen in Figure 2.1(a). If node A is transmitting to node B, a nearby node C might be outside the transmission range of node A and unaware of the ongoing communication. As a result, node C might initiate a transmission to node B simultaneously, causing a collision at node B despite both A and C following the CSMA protocol. This scenario illustrates that nodes cannot always detect ongoing transmissions from all other nodes, leading to collisions.

When collisions occur, transmitting nodes implement collision avoidance (CA) mechanisms such as request-to-send or clear-to-send (RTS) (CTS) handshake or increasing the random interval before retransmitting. While this approach reduces the likelihood of repeated collisions, it does not completely resolve the issue, especially in dense networks where the probability of collisions is higher.

An additional related CA vulnerability is the exposed node or terminal problem. In Figure 2.1(b), Node E tries to send data to Node D while Node F is transmitting a message to Node G. Since Node D is outside Node F's radio range, node E and Node F could theoretically transmit simultaneously without risk of collision. However, CSMA-CA prevents node E from transmitting because it detects node F's signal within its own radio range. Consequently, the Clear Channel Assessment (CCA) performed by node E will interpret the channel as busy during node F's transmission. The unneeded prevention is called the exposed node problem. The exposed node problem can be addressed in several ways, including repositioning nodes, reducing their output power to the minimum necessary for stable communication, and employing the RTS/CTS handshake mechanism.

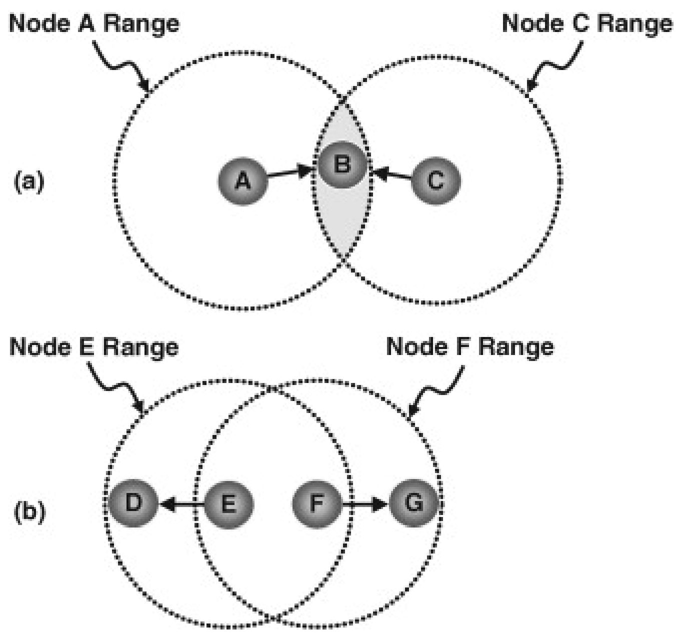


Figure 2.1: (a) The Hidden Node Problem and (b) The Exposed Node Problem. [17]

In summary, wireless communication channels in sensor networks face challenges such as transmission collisions and hidden nodes. While protocols like CSMA help mitigate some of these issues, they are not entirely sufficient. Additional strategies and protocols like RTS and CTS are used to mitigate issues and are necessary to enhance reliability and efficiency in wireless sensor networks.

2.3.3. Propagation

Fresnel Zone

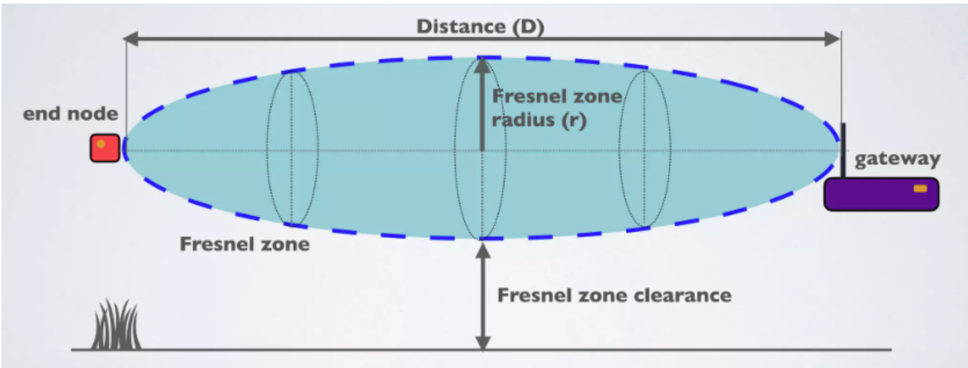


Figure 2.2: Fresnel Zone [18]

The Fresnel Zone is an ellipsoidal-shaped ring that spans the area of space around the direct line of sight between a transmitter and receiver. Its purpose is to define a region around the direct line of sight of a transmission that should be free of obstacles to prevent signal distortion. The maximum radius of this zone seen in Figure 2.2 can be defined by Equation 2.1. At 2.4 GHz frequency with a distance of less than 200 meters, the Fresnel Zone radius is less than 2.5 meters.

$$\text{Radius (m)} = 17.31 \times \sqrt{\frac{D\,(\text{km})}{4 \times f\,(\text{GHz})}} \tag{2.1}$$

Path Loss

Path loss can be described by the attenuation of a signal as it propagates through a medium. It is primarily used to describe the spreading of signal energy over increasing distances. At 2.4 GHz, radio waves mostly travel in straight lines, and the power of the signal decreases significantly with distance, typically by 6 dB with each doubling of the distance in free space. [16] This frequency is vulnerable to absorption by obstacles and environmental factors like rain. The 2.4 GHz band experiences considerable interference from various devices operating in the same spectrum, such as WiFi and Bluetooth, as well as electrical equipment like motors. [19]

The expression for free space path loss is detailed in Equation 2.2. Where P_r is received power, P_t is transmitted power(dBm), f is frequency(Hz), d distance(m) and c is the speed of light(m/s). It should be noted that the calculation for path loss through terrain is more complex than this and dependant on a variety of different factors. [20]

$$P_r = P_t - 20 \cdot \log\left(\frac{4\pi f d}{c}\right) \quad \text{[dBm]} \tag{2.2}$$

2.4. Protocol Architecture

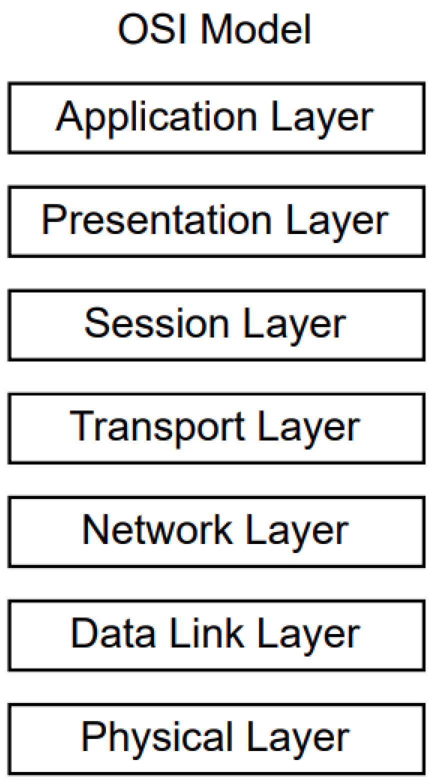


Figure 2.3: OSI Model [3]

2.4.1. Physical Layer

The physical layer is the lowest level of the Open Systems Interconnection Reference Model or OSI Model which is used to explain network function.

The Physical Layer is an important aspect of WSNs as this can determine many factors for a network such as network range or rate of data transfer. This layer is used to transmit the raw data bit stream over a medium.

IEEE802.11

Direct Sequence Spread Spectrum (DSSS) is a transmission method that merges a data signal and a high data rate bit sequence. This is called a chipping code. A chipping code is used to represent data at a much higher frequency for transmission. It is decoded after it has been received. DSSS is used to manage interference and susceptibility to interference in IEEE 802.11 standard [21]. IEEE802.11n is the standard that will be focused on in this work. It operates in the 2.4 GHz Industrial band.

2.4.2. Data link Layer

The Data Link layer is responsible for the transfer of data packets between directly connected nodes. It sorts the actual data into manageable packets for transmission with each packet including control information like source and destination addresses, sequence numbers and error checking codes. The structure provided by this layer informs the receiver about the data and how to process it.

Error Detection

This layer is able to implement error detection mechanisms to validate the received data and check data corruption. Methods such as checksum validation and Cyclic Redundancy Check (CRC) can identify errors that occur during transmission due to noise, interference or fading. Checksums use a simple method where the sender generates a checksum value based on the data and sends it with the data. The receiver recalculates the checksum and compares it with the original from the sender. CRC is a more robust detection method that uses polynomial division to generate a unique checksum for the data.

2.4.3. Medium Access Layer

The MAC layer regulates how nodes share the same communication channel in a network. This is especially important to manage collisions and ensure reliable communications in a network. The MAC layer's aim is to ensure shared access to the communication medium to avoid collisions, ensure fair usage, and optimise network performance.

Aloha

One of the earliest protocols designed for random access to a shared communication medium operates on a simple principle. A node will transmit data whenever they have data available to send. If a node does not receive an acknowledgement for the transmission and thus detects a collision, it waits for a random time before attempting to send the data again. The maximum theoretical throughput for this method of Pure ALOHA is 18.4%. [22]

CSMA

CSMA evolved from ALOHA to provide a more structured approach to medium access, incorporating the principle of sensing the channel before transmitting. However, this approach requires the sender to have hardware capable of detecting transmissions on the communication channel. Nodes first listen to the channel to determine if it is idle before transmitting. If the channel is busy the node will wait until it becomes free before sending data. When sensing an idle channel, nodes may still experience a collision if another node begins transmitting simultaneously.

CSMA/CA is a common protocol used in wireless networks, where nodes cannot detect collisions. Instead, nodes use techniques like RTS/CTS to minimise the chance of collisions by reserving the channel before data transmission.

2.4.4. Network Layer

The Network Layer in WSNs is responsible for routing data from sensor nodes to the BS or gateway. Due to the characteristics of WSNs such as the limited battery life of sensor nodes and the dynamic nature of the network topology, efficient routing strategies can help ensure reliable data transmission while minimising energy consumption.

Routing

Routing in WSNs uses various strategies designed to optimise data flow in a network. These strategies determine where data is transmitted around a network. Protocols can be proactive, maintaining up-to-date routing tables of relevant network information, or reactive and establish routes on demand.

Flooding is one of the simplest algorithms used in routing. When a source node has data to send, it broadcasts the data packet to all its neighbours. Each node receiving the packet forwards it to all its neighbouring nodes, except for the node from which it received the packet. This method can improve the chance for data to reach the destination. However, it has excessive redundancy and creates lots of traffic volume and potential for congestion.

Distance Vector Routing is a traditional routing algorithm where each node maintains a table of the best-known costs to every other node in the network. Nodes periodically share their routing tables with their neighbours, allowing them to update their own tables based on the information received. This is a simple and adaptable topology that is capable of handling larger networks, yet it can have lower latency.

2.4.5. RSSI

The received signal strength indicator provides a signal strength measurement by the receiver of a transmission. From Katircioğlu et al. "In embedded devices it is common to define RSSI as a ratio of a received power to a reference power", Equation 2.3. [20]

$$\mathrm{RSSI} = 10\log_{10}\left(\frac{P_{\mathrm{RX}}}{P_{\mathrm{Ref}}}\right) \tag{2.3}$$

2.4.6. ESPNOW

ESPNOW is a network stack protocol primarily used on IoT devices. It simplifies the OSI model as seen in Figure 2.4. The protocol uses the standard IEEE802.11 DSSS modulation [21] with a bit rate of 1 Mbit/s. It makes use of the IEEE802.11 CSMA standard for collision avoidance and has 2 modes of communication: unicast and broadcast.

When using unicast mode, the sender communicates directly with a single receiver by sending a data packet that specifies the receiver's MAC address in the packet header. The sender expects to receive an acknowledgement (ACK) from the receiver after the transmission. If the ACK is not received within a certain time, the sender will retransmit the packet.

When using Broadcast mode, the sender sends a data packet to all receiver's devices at once by using a specific broadcast MAC address in the packet header. Unlike unicast transmissions, the sender does not require ACK from any of the receivers. [3]

The data packet format can be seen in Figure 2.5. The protocol has an overhead of 43 bytes, with a 0-250 byte body allowance.

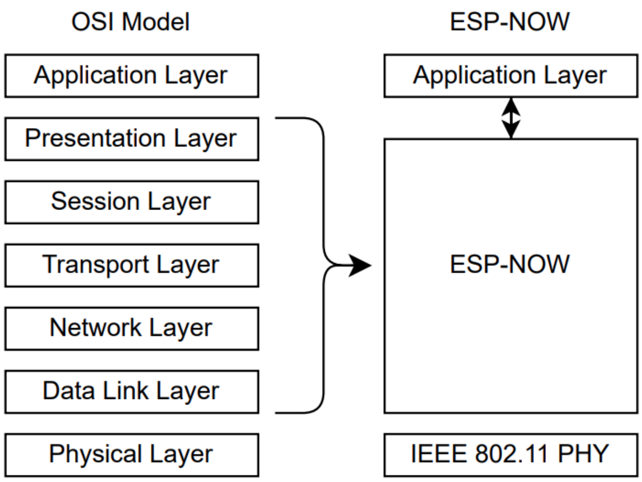


Figure 2.4: ESPNOW protocol stack [3]

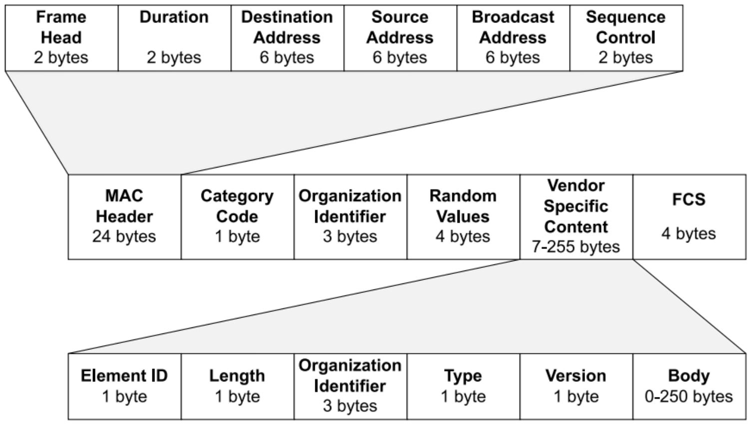


Figure 2.5: ESPNOW data packet structure format [3]

Chapter Summary

This chapter has covered the applications for WSNs in Smart Agriculture focused on vineyard settings and routing optimisation in this area. The next chapter will look at the design of the system for this project.

Chapter 3

System Design

3.1. System Overview

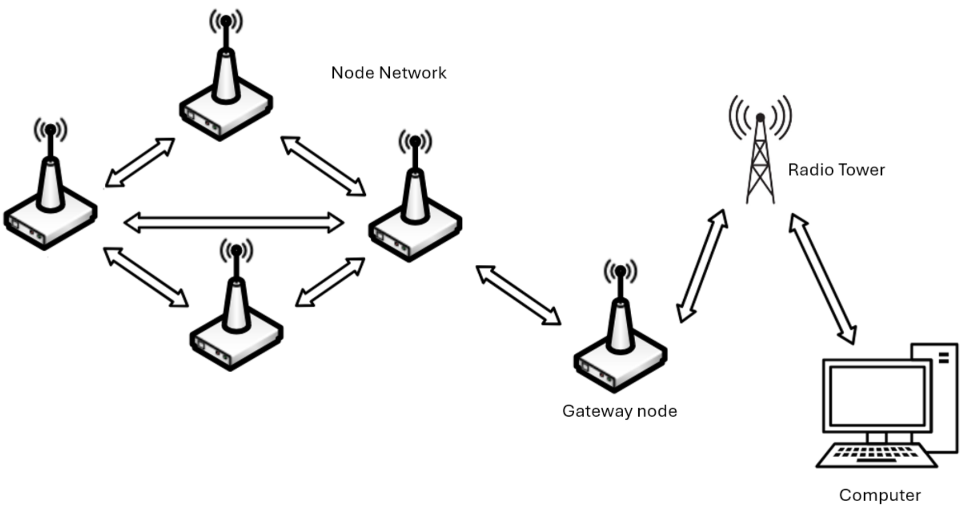


Figure 3.1: System Diagram

3.2. Introduction

The selection choices for the hardware in this project were made to provide a platform to develop a routing algorithm with emphasis on energy efficiency. The hardware modules chosen do not necessarily represent the best choice for a system that requires a low power consumption. The decision-making process prioritised components that align with the project's objectives and budget constraints. The use of pre-made modules was a choice to streamline the design process and minimise lead time, enabling quicker progress in developing the routing algorithm.

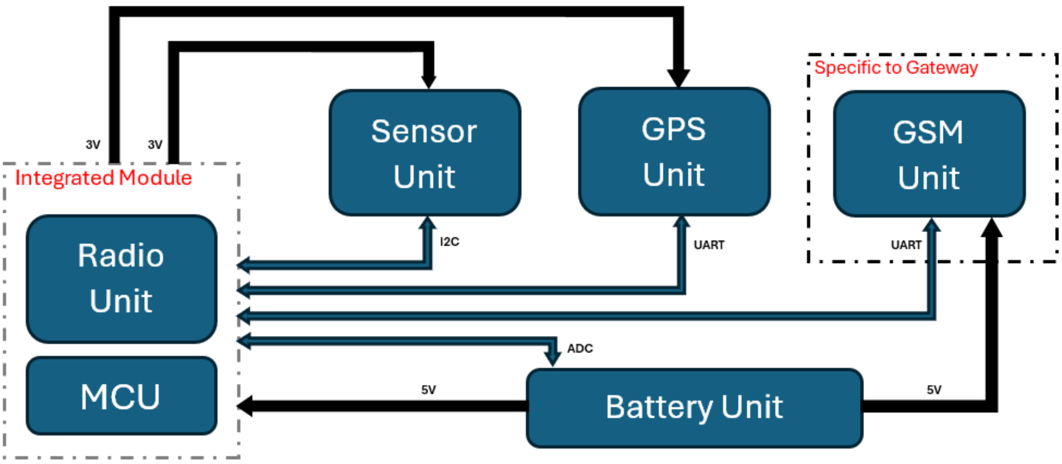


Figure 3.2: Node System Diagram

3.3. Hardware Choices

3.3.1. Micro-controller Unit

The foundation of all the nodes in the system is the MCU. This will process the data collected from the peripherals and make the routing decisions to forward and send data through the network. It should have a RTC and sufficient memory to process the payloads it must generate and forward to the network.

ESP32-WROOM-DA

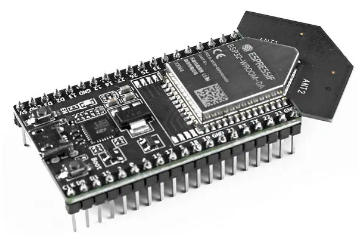


Figure 3.3: ESP32-WROOM-DA Development board

This module was chosen for the project as it eliminates the need for an additional radio module and provides a platform to build the proposed routing protocols on the MCU at a low cost. A high quality RTC is present and no additional memory will be needed on the unit. It can also run at relatively low power and has the ability to run a server on an open port.

3.3.2. GPS Unit

MKS ATGM332D 5N31 GPS

The MKS ATGM332D 5N31 GPS module was selected for its high performance and accuracy. It has low power consumption with a maximum positional error of 2.5 meters, making it an excellent choice for the location-based application of this project within the budget constraints.

3.3.3. Sensor Unit

BME280

The BME280 sensor was chosen for its compact size, affordability, and accuracy in environmental measurements. It offers low power consumption while providing a data package for the network at a low cost, making it good for monitoring environmental conditions.

3.3.4. GSM Unit

800C GSM Module

The 800C GSM module was selected for its ability to send reliable communication over cellular networks. Its features and low power consumption make it suitable for transmission of the data package from the gateway node.

3.3.5. Power Supply

Samsung 18650 3500mAh 3.7V Lion Battery

This battery provides a reliable power source for the system, offering a good balance between capacity and size, essential for long operations in remote locations.

Single Battery Shield 18650

The Single Battery Shield 18650 was chosen to manage the power supply efficiently, ensuring safe operation of the battery and charging. It offers easy integration with the rest of the system.

3.4. Metrics

3.4.1. Performance Measurement Objectives

To evaluate the effectiveness and efficiency of the proposed routing algorithm for the smart agriculture sensor network, the following key metrics will be measured:

Node Energy Consumption:

The energy consumed by each node during a single data-gathering sequence, providing insight into the efficiency of energy usage for individual operations.

Quality of Service

For the physical implementation quality of service will be evaluated by comparing the number of data requests sent and the number of data replies received. This metric is not applicable to the simulation as the messages will not be lost due to unforeseen circumstances.

Routing Efficiency:

These metrics will be the primary measure of network performance:

- *Average Transmission Hop Count:* The average number of hops required for transmissions to reach their destination, indicating the effectiveness of the routing protocol.
- *Average Latency:* The average time taken for a packet to be delivered from the source to the destination, providing insight into the responsiveness of the network.
- *Message count:* The number of data request and reply messages received at each node. This can provide insight into how the node routed the data and spread the load across the network.

Scalability:

Evaluation of the network's performance as the number of nodes increase assesses how well the routing algorithm can adapt to larger scales.

Topology Changes:

The ability of the routing algorithm to handle changes in network topology, such as node exclusions, additions or failures, displays robust communication under dynamic conditions.

3.4.2. Method to achieve Objectives

To achieve these objectives the following approaches will be employed:

- **Energy Measurement Tools:** Nodes will be measured to determine the average power consumed by a data-gathering sequence.
- **Simulation Environment:** A simulation framework will be implemented to model the sensor network, allowing for the testing of routing protocols under various conditions and facilitating the collection of data on energy consumption, latency, and hop count.
- **Data Logging:** Implement logging mechanisms to record energy usage, packet delivery times, and routing information throughout the operation of the network.
- **Scalability Testing:** Increase the number of nodes in the simulation and observe the network's performance metrics, including latency and packet delivery ratio.
- **Topology Management:** Introduce dynamic scenarios in the simulation where nodes are randomly added or removed to evaluate the adaptability of the routing algorithm.
- **Load Testing:** Simulate different network traffic conditions by varying the number of packets generated by nodes, assessing how the routing protocol manages under increased load.

Summary

By measuring these metrics and using the outlined methods, the project will evaluate the performance and energy efficiency of the proposed routing protocol for smart agriculture WSN's. The next chapter will focus on the detailed design of the system.

Chapter 4

Detailed Design

This chapter covers the detailed design process that was used to build the full network of nodes. It investigates the hardware design needed for the modules used, the PCB design process and steps followed by the Software design of the simulation and physical system.

4.1. Hardware Design

4.1.1. MCU

The ESP32-WROOM-DA module is used on each node in the system. It is flashed with Micropython version 1.23.0 firmware which is used to run node operations. An ADC pin was used to sample the battery voltage and communication buses are configured to use UART and I2C serial communications. This module is used to provide a 3.3V power supply to the Sensor Unit and the GPS Unit.

4.1.2. Battery Unit

A Samsung 18650 3500mAh 3.7V Lion Battery is used in with a Battery Shield to provide power to the nodes. The module outputs 5V 2A power supply to the MCU and GSM Unit. An additional connection is made to the positive battery terminal to measure the battery voltage. The MCU pins are rated for 3.3V so a voltage divider is designed to bring the battery voltage into this range. Using Equation 4.1, suitable values were chosen for the divider with an acceptable percentage error, as seen in Equation 4.2, and power loss calculated using Equation 4.3.

$$V_{out} = V_{in} \times \frac{R2}{R1 + R2} \approx 2.995\,\mathrm{V} \tag{4.1}$$

$$\text{Error} = \left(\frac{3.0 - 2.9953}{3.0}\right) \times 100 \approx 0.174\% \tag{4.2}$$

$$P = \frac{V_{in}^2}{R_{total}} \approx 153.391\,\mu\mathrm{W} \tag{4.3}$$

4.1.3. GSM Unit

The 800C GSM Module requires a dedicated 5V 2A power supply with decoupling capacitors on the supply line. The recommended values are 100uF and 10uF for normal operation.

4.2. PCB Design

The PCB boards were designed using Altium Designer. A total of five boards were made, one for each node. A module library was created for each module by physically measuring the dimensions and creating schematics. The full system schematic can be seen in Figure 4.1. The schematic connections were mapped to a single sided PCB and the resulting board was generated as seen in Figure 4.2.

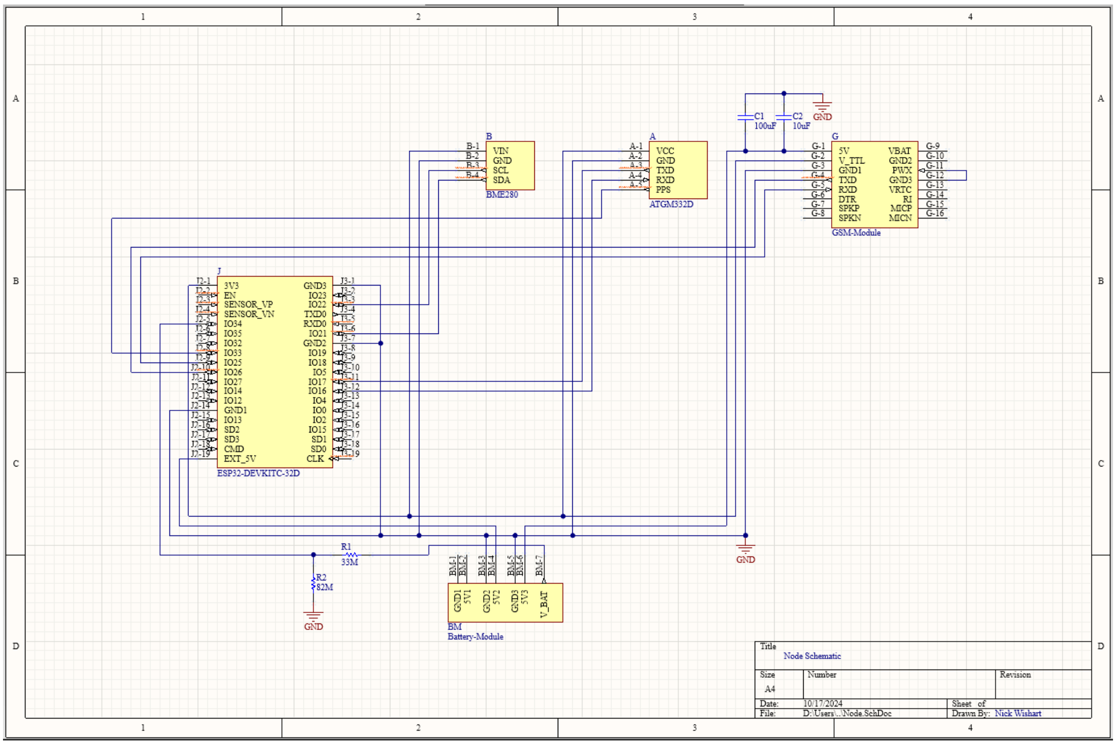


Figure 4.1: PCB Schematic Diagram

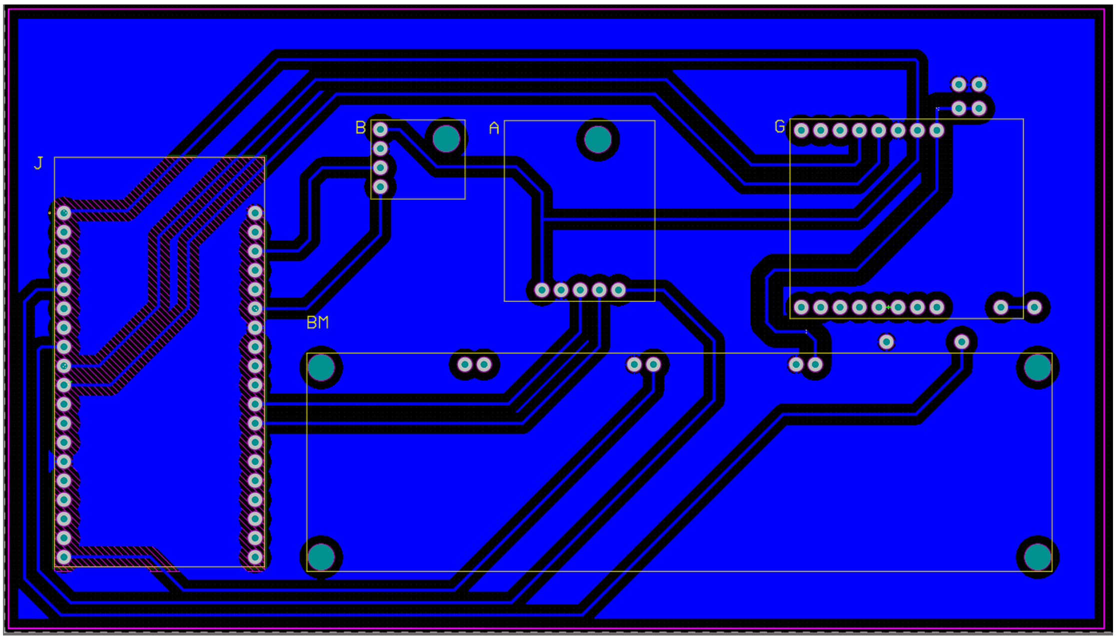


Figure 4.2: PCB Board

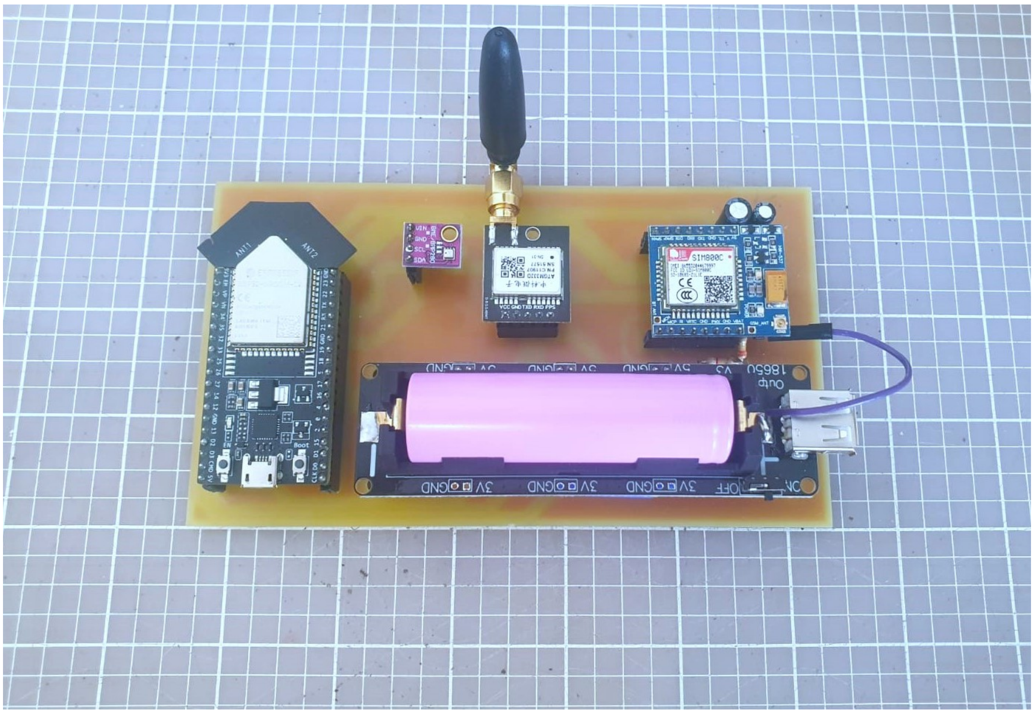


Figure 4.3: Assembled Node

4.3. Software Design

This section will cover the software development of the WSN system used in this project. It will focus on the node operations and action flow. Furthermore, it will discuss the format of the data logs that are output by the gateway node. For this project, two separate code-bases were created, one for the physical five node implementation and one for the simulated network.

4.3.1. Physical

The physical implementation uses MicroPython, a version of Python3 designed for constrained environments such as microcontrollers. The microcontroller performs various tasks including sensor data collection (from GPS and BME280), battery monitoring, routing decisions based on network conditions (RSSI, battery levels), and data transmission using the ESP-NOW library for communication between nodes.

A Node runs a python script that manages the operations of the microcontroller. The breakdown of its functionality is as follows:

Initialisation

The Node imports a configuration file that provides WiFi connection credentials to a hotspot for wireless log output monitoring. Various libraries are imported to handle communication, networking, timing, random delays, and message encoding/decoding. Microcontroller specific modules are imported to manage hardware peripherals such as I2C for sensors and UART for GPS/GSM modules. Global constants including the MAC address of the network gateway, timeouts, delays, communication intervals and the weight factor are used in calculating overheads for routing decisions.

The Node class represents a node in the network. Each node can act as a regular sensor node or a gateway, depending on its MAC address. A Node begins by initialising ESP-NOW and dynamically determines its own mac address for peer-to-peer communication. It then attempts to connect to the WiFi hotspot defined in the configuration file before starting a Telnet server for remote terminal access.

After the network initilisation the sensors and peripherals are configured. The I2C interface is set up to communicate with the BME280 environmental sensor. The I2C bus is initialised with the appropriate pins and clock frequency, ensuring proper communication with the sensor. The node sets up an ADC pin to monitor the battery voltage. Next, the UART interface is set up for both the GPS and GSM modules.

The GPS module is used on all nodes to update the geographical position data as well as the local clock data of the node. It then uses the PPS signal to retain synchronisation even if a GPS fix is lost.

The GSM module, only active on the gateway node, facilitates communication with the central server or cloud for remote data transmission. This module allows the gateway to send collected data from the entire network to a remote monitoring system, enabling real-time updates without relying on local storage. The Node Initialisation sequence can be seen in Figure 4.4.

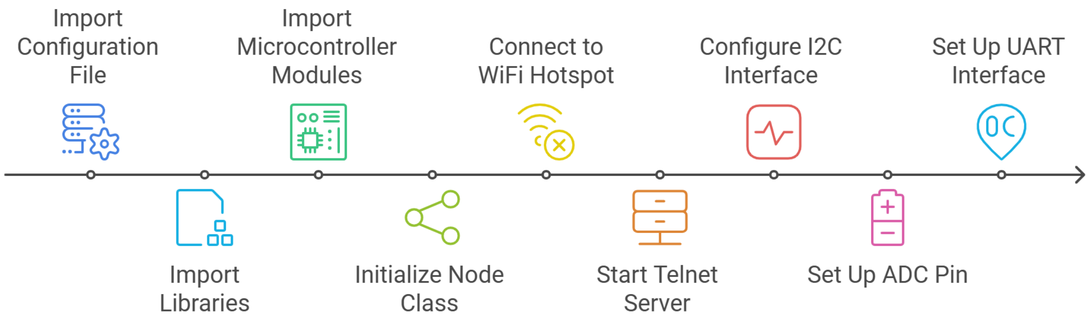


Figure 4.4: Node Initialization and Configuration Sequence

It should be noted that after the the functionality of the GSM module was verified it was no longer used as the log output mechanism for this project due to the extra costs involved(sim card) with its operation. Instead the logs were transmitted to a laptop connected to the gateway node via the telnet server. After initialising the sensors and peripherals, the node enters the main communication loop.

Design Considerations

The operation of the network routing was investigated and tested during development. Initially a solution that used delayed execution of functions and timing was implemented. However, this presented challenges relating to the synchronisation of the nodes. It was found that there was a very small margin for error with the synchronisation. Instead of this method an action/reaction method was developed to alleviate these concerns.

The main loop handles the operation of all the nodes in the system. The network was designed to span distances greater than the transmission distance of one singular node. Thus a simple method of operation was chosen for network communication.

Each node has two types of transmissions, a data request 'dreq' and a data reply 'drep'. The data request is a broadcast to all surrounding nodes in reception range. It performs the function of node discovery in the network by sending a data packet with the required information used to populate the neighbour tables of the surrounding nodes. The data reply transmission is a unicast message containing the sensor payload and routing metrics, sent back through the network towards the gateway node.

Included in the data request transmission is the node specific information used to calculate the overhead or cost to transmit back to the gateway node, the overhead itself and the current path to the gateway. The overhead is calculated using accumulated values of battery level and RSSI. These values are normalised and weighted in the overhead calculation.

A struct was used to pack the data before being sent over ESPNOW which has a size limit of 250 bytes per message. The compression significantly reduces the overhead compared to sending raw data, but it is not optimised and results in a 25 byte string which is sufficient for this purpose. The packed data includes all relevant sensor readings, including the MAC address, timestamp, GPS coordinates, environmental data (temperature, humidity, pressure), battery level, and communication metrics like RSSI and hop count.

Main Loop

All nodes begin the main loop idle, listening for a transmission, except for the gateway node, which initiates the data acquisition sequence by sending a single data request transmission. On receiving a data request message, a node will update its neighbour table with the newly acquired information, taking note of the address of the node which lies in its path back to the gateway, and then broadcast a data request message of its own.

If a node receives a data request from another node offering a lower overhead than an existing connection, the path to the gateway is updated with the new node. The node overhead is recalculated and a new data request is broadcasted with the updated information.

After a delay, to allow any node updates to take place, a node will check to see if its an edge node. An edge node is defined as not having any neighbours in its neighbour table with its own address in their path. In other words, none of the surrounding nodes will choose this node to forward their data onto the gateway node. Therefore, it must initiate a data reply.

On receiving a data reply, a node will record and add its own data to a data cache if it has not already done so, and increment the hop count of the received data packet before adding it to its data cache. The node will then wait for all other nodes in its neighbour table that have its address in their path before sending its own data reply to the next node.

Once the data reply reaches the gateway, the gateway information is added to the data packet and log files are generated or updated with the sequence data. The data acquisition sequence can occur indefinitely or for a predetermined number of rounds.

4.3.2. Simulation

In addition to the physical system, a simulation was developed to thoroughly evaluate the implemented routing algorithm. In the early stages of development it was decided to use simpy [23], a python library specifically chosen for its discrete time environment. Another library, wsnsimpy [24] was used to help visualise the system.

The simulation was built using the same methods as the physical implementation with a few changes. These differences consist of the sensor data and the positions of the nodes being randomly generated. The simulation is able to increase the node count of the network enabling scalability analysis. Figures 4.5, 4.6, 4.7 show the general operation of the simulation network system.

The code for the simulation implementation[1] and the physical implementation[2] can be found in their github repositories.

Chapter Summary

This chapter has covered the detailed design of the project. The next chapter will look into the results achieved from testing these systems for the project.

[1] https://github.com/nWish8/Sim
[2] https://github.com/nWish8/DevNet

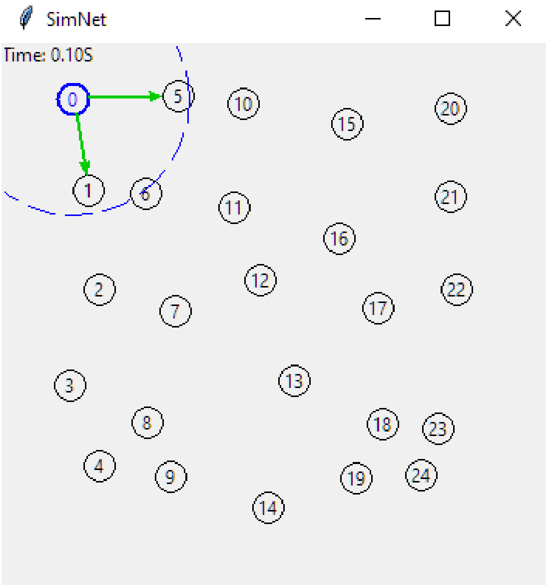


Figure 4.5: Data request (green arrow) from gateway node

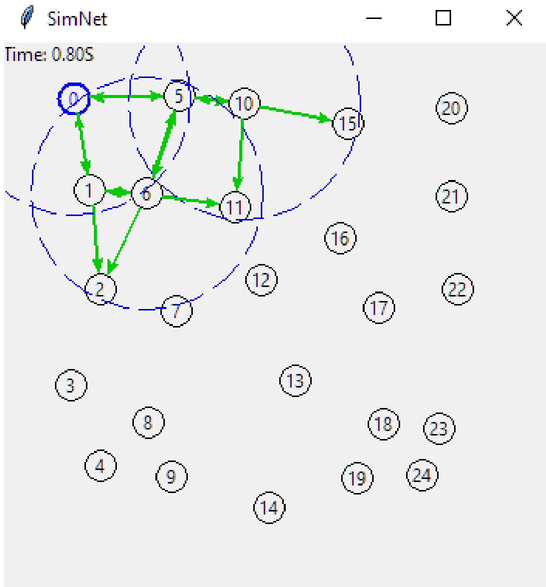


Figure 4.6: Network discovery

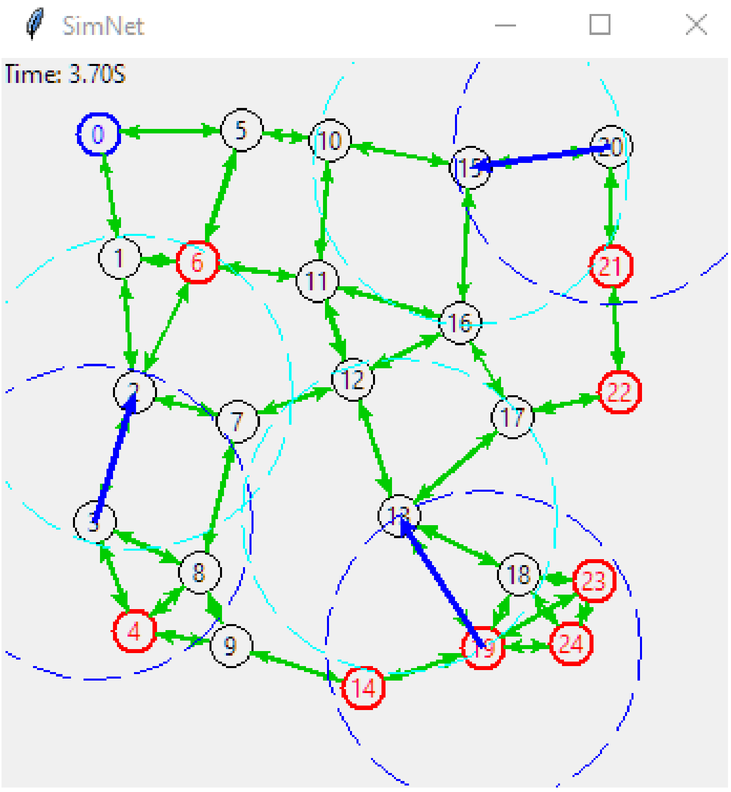


Figure 4.7: Edge Node discovery(red), Data reply(blue arrow) and message ACK(cyan ring)

Chapter 5

Results

This chapter will document the results acquired by the experiments conducted in this project. It will look at the transmission power characteristics, the vineyard test physical and simulated results, and then finish with scalability tests.

5.1. Transmission Power

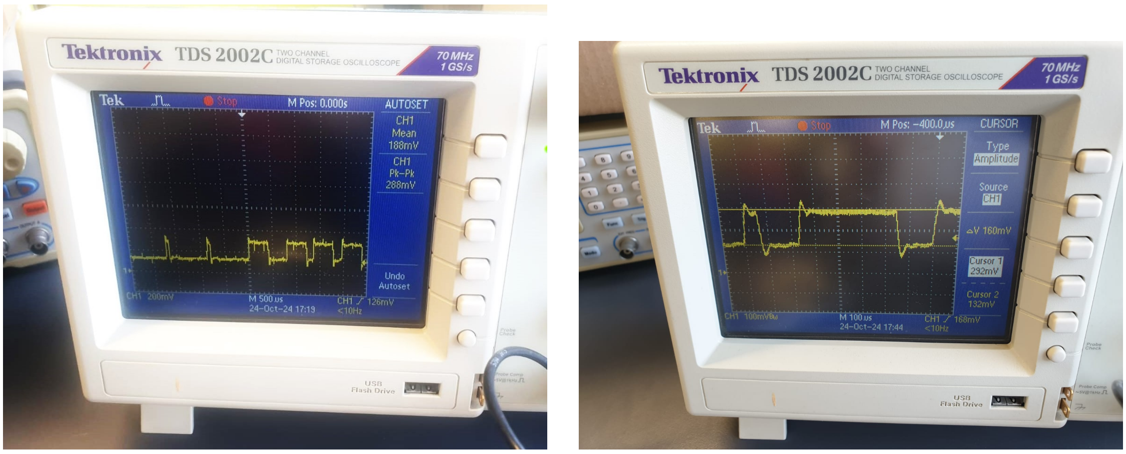
The transmission power of a node was measured to determine the consumption of power for a single transmission. A shunt resistor was placed in the series with the battery supply, the ESP32-WROOM-DA module and the voltage measured across it. The RMS current and power for a single transmission was calculated. This value includes the power consumption of a signal transmit and any computation that occurs in a sequence. The results can be seen in Figures 5.1, 5.2 and Table 5.1

Parameter	Value
Calculated Current	162 mA
Calculated Power Consumption	0.81 W
Sequence Duration	160 ms

Table 5.1: Power Characteristics of a Single Transmission Sequence

(a) Full Tx Sequence Time (b) Full Tx Sequence Voltage

Figure 5.1: Oscilloscope Captures: (a) Full Sequence Time information, (b) Full Sequence Voltage information



(c) Tx Time (d) Tx Voltage

Figure 5.2: Oscilloscope Captures: (c) Tx Sequence Time waveform, (d) Tx Sequence Voltage characteristics

5.2. Vineyard Test

A test was conducted in a vineyard to evaluate the performance of the system. After an initial trial, all the nodes were able to transmit directly to the gateway across the whole vineyard. This did not demonstrate the routing algorithm and thus the ESP32 transmit power was adjusted from its maximum setting of 19.5 dBm to a value of 5 dBm. This ensures the routing algorithm is not bypassed and all the nodes are not in range of the gateway node.

5.2.1. Physical

The nodes were placed around the vineyard at suitable distances from the gateway to analyse the performance of the routing algorithm. The test recorded metrics from all the nodes such as quality of service, average latency, voltage drop of the battery over 100 sequences, average RSSI received by each node and the average reply message hop count for each node. The node placement can be seen in Figure 5.3. For 100 data request transmissions the results can be seen in Table 5.2 and Table 5.3.

Sequences	QoS (%)	Avg. Latency (s)
100	73.4	2.247

Table 5.2: QoS and Latency Data over 100 sequences

Node	Voltage (V)	Battery (%)	RSSI	Hop Count
Gateway	0.1	91.67	0	0
1	0.1	91.67	-88.7	1.1
2	0.2	83.33	-86.8	1.6
3	0.1	91.67	-89.9	2.6
4	0.1	91.67	-90.2	2.6

Table 5.3: Voltage drop of the battery, battery consumption, average RSSI received by a node, and average hop count data over 100 sequences

Figure 5.3: Vineyard test site

Node	PosX	PosY	Temp	Hum	Press
Gateway	-33.95073	18.86967	25.71	27.14	996.3
1	-33.95036	18.86832	34.0113	18.77	985.4
2	-33.94979	18.86729	33.1483	20.97	988.0
3	-33.94909	18.86689	32.3386	19.30	986.0
4	-33.94864	18.86760	24.5375	30.00	1000.5

Table 5.4: GPS Position and Average Temperature, Humidity, and Pressure Data over 100 Sequences

5.2.2. Simulation

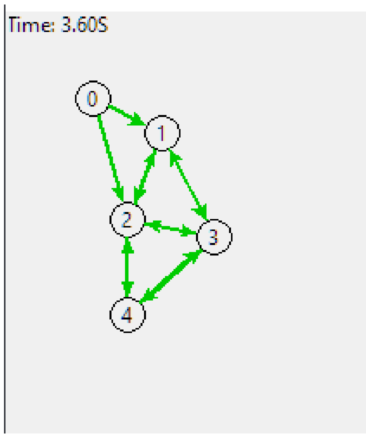


Figure 5.4: Simulation Layout

A simulated version of the vineyard test was conducted with fixed node placement to mimic the physical implementation for 100 data request transmissions. The layout is shown in Figure 5.4. The results can be seen in Table 5.2 and Table 5.6.

Sequences	Avg. Latency (s)
100	2.716

Table 5.5: Latency Data over 100 sequences

Node	Battery (%)	RSSI	Hop Count	Rx 'DREQ'	Rx 'DREP'
Gateway	94.6	0.0	0	200	202
1	89.9	-73.9	1	302	46
2	90.5	-120.4	1	404	158
3	87.3	-84.3	2	302	2
4	91.8	-90.9	2	201	0

Table 5.6: Remaining battery percentage, average RSSI received at node, average hop count, and number of message type received at node over 100 sequences

A graph comparing the physical and simulated average hop count values can be seen in Figure 5.5. The number and type of messages received by each node are shown in Figure 5.6.

These results show that the simulation modelled the physical system well and slight differences in the battery percentage, RSSI, and hop count can be attributed to environmental conditions, interference and path loss in the physical testing. The percentage error is an acceptable level, proving the efficacy of the simulation developed.

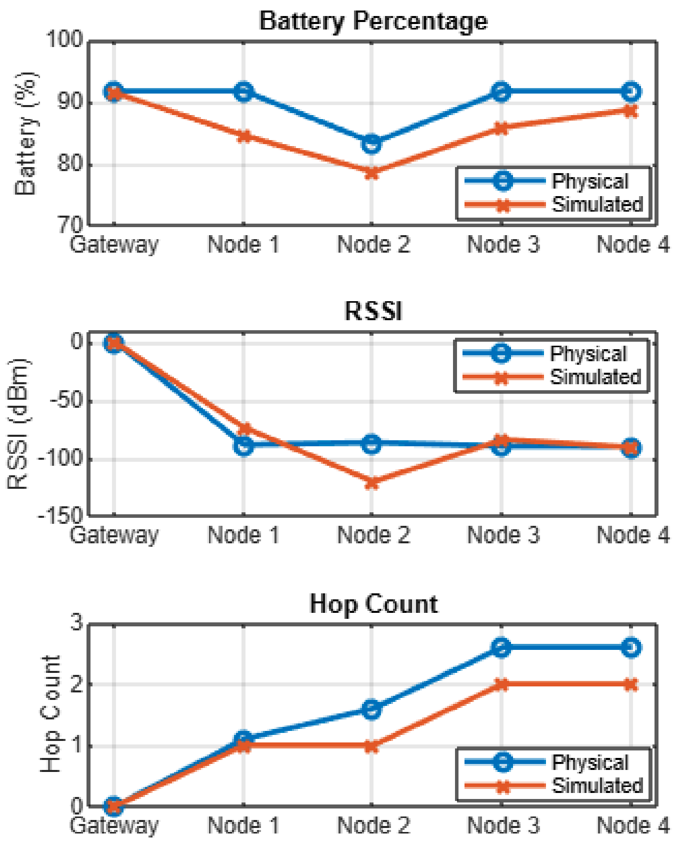


Figure 5.5: Vineyard test metrics

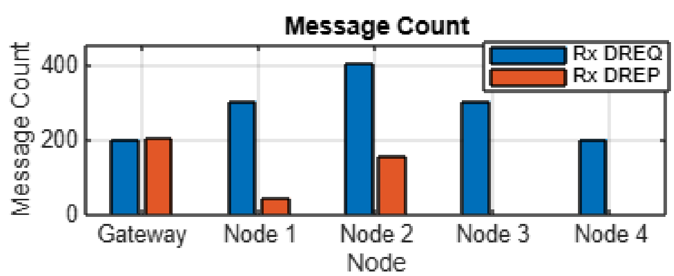


Figure 5.6: Vineyard test received message count

5.3. Scalability

A simulation was run for various increased node counts in the network to determine performance under load. The results of the metrics can be seen in Table 5.7 and Figure 5.7. The results indicate that system latency initially increases with a higher node count, but this effect gradually diminishes as the network expands, stabilizing as this number increases. As expected, the hop count rises with node count, with the increased distance of edge nodes from the gateway in larger systems. The average battery percentage across tests shows the system's ability to effectively distribute network load, preventing excessive drain on individual nodes and avoiding over-reliance on critical pathways.

Node Count	Avg. Latency (s)	Avg. Hop Count
25	6.4	4.01
50	8.15	5.102
100	12.2	8.72

Table 5.7: Latency and Hop Count Data over 50 sequences with various node counts

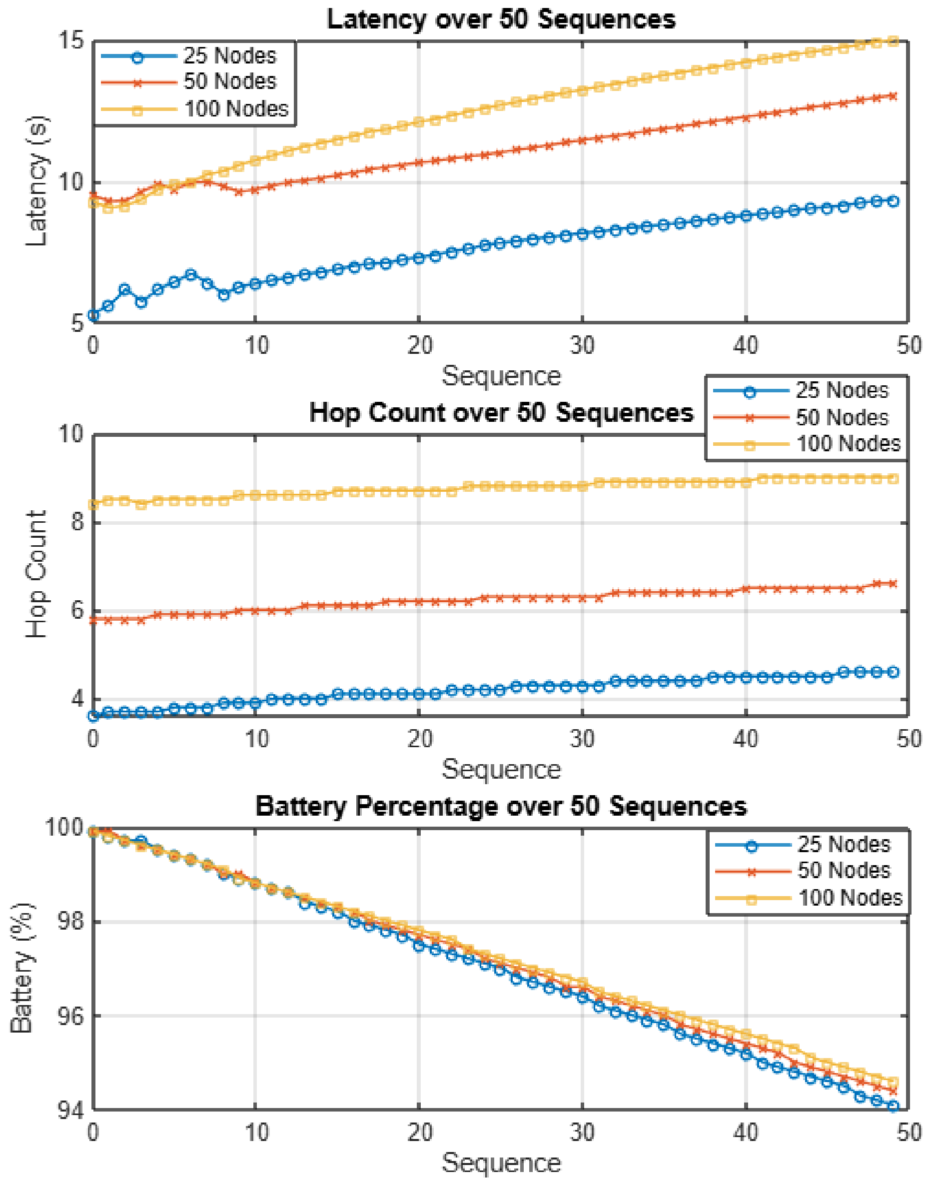


Figure 5.7: Average load test metrics for 25, 50, and 100 nodes over 50 sequences

Figure 5.8 shows a comparison of DREQ and DREP message counts for the scalability tests. It was determined from the results that the number of messages received by each node has a direct correlation with the power consumption of that node. Edge nodes received a significantly lower number of DREQ messages while nodes on a critical path to the gateway received an increased number of DREP messages.

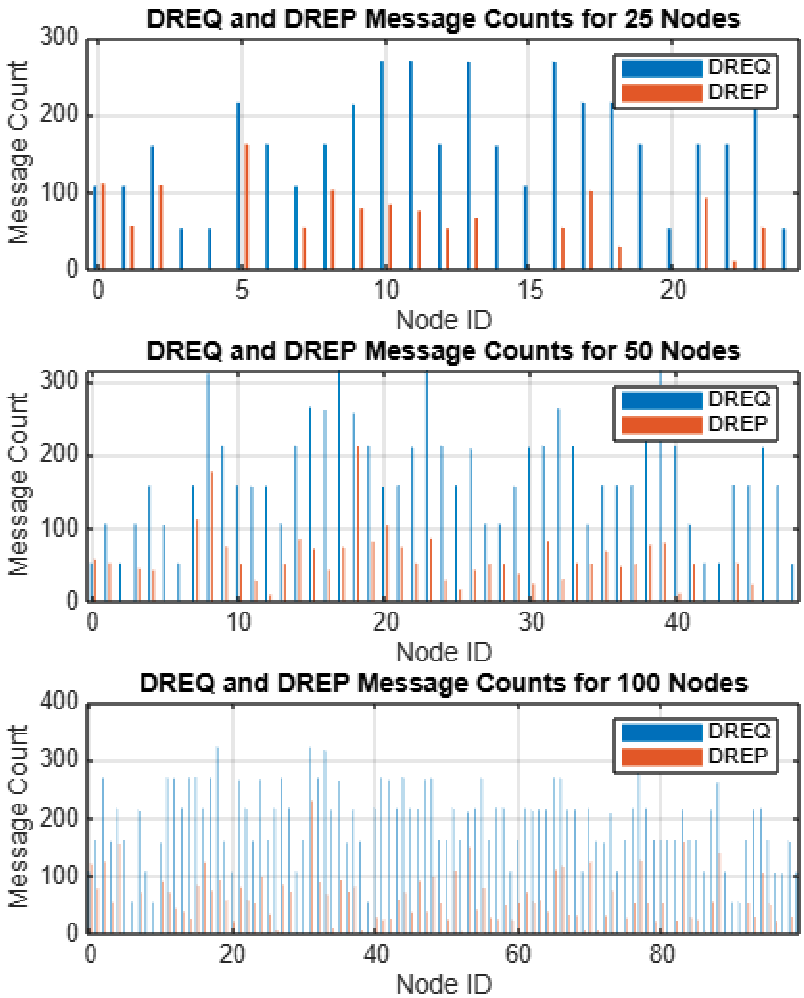
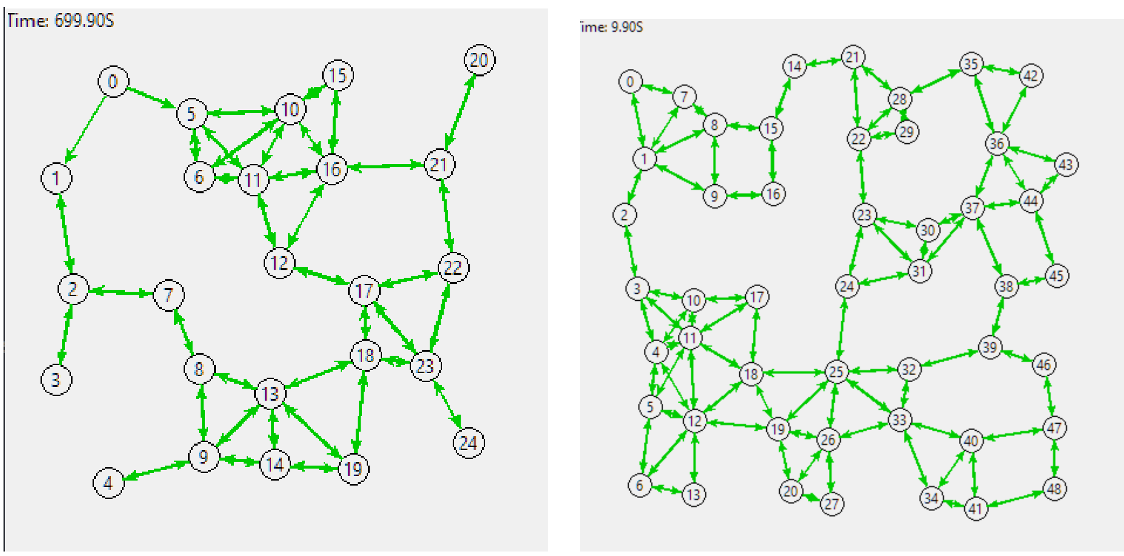


Figure 5.8: Comparison of DREQ and DREP Message Counts for 25, 50, and 100 Nodes over 50 sequences



(a) 25 Nodes (b) 50 Nodes

Figure 5.9: Network Simulations for: (a) 25 Nodes, (b) 50 Nodes

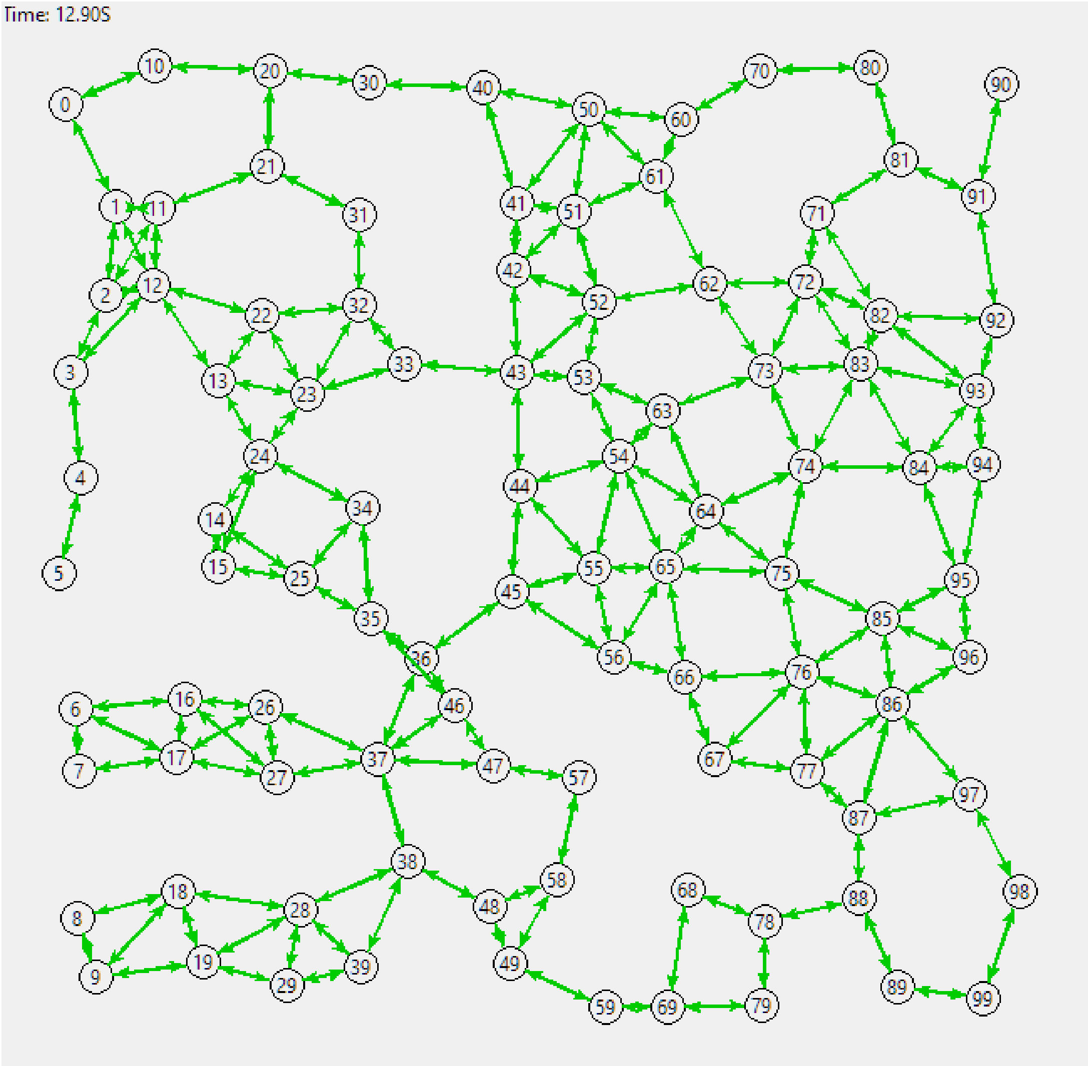


Figure 5.10: Network Simulation for 100 Nodes

Chapter Summary

This chapter has covered the results achieved from testing the systems and discussed the outcome of the metrics analysed. The next chapter will contain the conclusions drawn from the results of the report.

Chapter 6

Summary and Conclusion

The project focused on developing an efficient communication and routing protocol tailored for a smart agriculture sensor network, particularly in a vineyard environment. The main objective was to design a network that could monitor key environmental factors, such as temperature and humidity while optimising energy consumption through message routing. By implementing an energy-aware routing protocol and embedded development boards, the network looks to improve communication efficiency and minimise power usage.

The project was divided into two main phases: simulation and physical implementation. During the simulation phase, a routing protocol was developed and network configurations were tested to evaluate their performance in terms of scalability, power efficiency, and reliability. In the physical implementation phase, a prototype was deployed in a vineyard to collect environmental data and validate the system's performance. Metrics such as node battery level, RSSI, hop count, and latency were collected and analysed.

The overall goal of optimising energy consumption while maintaining robust data transmission was achieved through the design of a custom routing algorithm, which was tested across various scenarios and network scalability tests with up to 100 nodes.

6.1. Conclusion

The ESP32 device proved capable in handling the processing of the system and showed no signs of performance loss in testing. The integrated radio device in the ESP32 module displayed signs of interference which are most likely due to the 2.4 GHz band being widely used. It performed well in its implementation, but it is not a likely choice for a commercial system as there are other more power-efficient units on the market.

The energy consumption per node was reduced by distributing the load of the network, optimising the routing algorithm and balancing the trade-off between communication reliability and power usage. The network was able to scale effectively, maintaining an acceptable hop count and latency as the number of nodes was increased.

The protocol proved to be adaptable to dynamic changes in the network topology, such as node failures or the addition of new nodes in both physical and simulated environments without any reduction in performance. The adaptability showcased by this protocol can be attributed to the entire network discovery occurring with each data request from the gateway node. This is a greedy method that uses a flooding technique to start every sequence in order to achieve this goal. This method could be modified to perform this operation at intervals however, this would be at the expense of adaptability.

6.2. Project Outcomes

This project successfully developed and tested a communication and routing protocol that is both energy-efficient and robust for use in smart agriculture networks. The results demonstrated that the protocol could handle node failures and topological changes, allowing continuous data transmission in challenging conditions. The energy-aware routing protocol showed improvements in energy consumption compared to simple routing methods, particularly in scenarios with a large number of nodes. The results showed that the routing protocol was able to balance the power load on a large network successfully.

It was found that power consumption is directly related to the number of messages received by the node. This is due to the functioning of the routing system. A message transmit is the most energy-intensive operation of a node, followed by a transmission receive and processing. The protocol instructs all nodes to send only one request and reply message, except for special cases. This means that the difference in energy consumption across the nodes is mainly made up of received transmissions and processing.

6.3. Recommendations

These findings show that the proposed routing protocol is well-suited for deployment in vineyard environments and other agricultural settings where power efficiency and reliability are important. Future work could explore further optimisation of the routing protocol for different environments and investigate additional sensor types to enhance the network's functionality. Furthermore, future research could also focus on optimising the protocol for even larger networks, incorporating variable transmit power based on RSSI and additional energy-harvesting techniques to further reduce the reliance on batteries. Additionally, exploring alternative communication protocols or integrating machine learning to optimise the calculation of the overhead for a more adaptive routing strategy could be considered.

Bibliography

[1] N. Nell, N. Mitton, T. Niesler, and R. Wolhuter, "An Ad-Hoc Sensor Network for Vineyard Monitoring," in *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, Spier, South Africa, Nov. 2021. [Online]. Available: https://inria.hal.science/hal-03347944

[2] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '01. New York, NY, USA: Association for Computing Machinery, Jul. 2001, p. 272–287. [Online]. Available: https://doi.org/10.1145/381677.381703

[3] D. Urazayev, A. Eduard, M. Ahsan, and D. Zorbas, "Indoor Performance Evaluation of ESP-NOW," *Nazarbayev University Journal*, May 2023.

[4] AFGRI, "Precision Agriculture." [Online]. Available: https://www.afgri.co.za/precision-agriculture/

[5] K. Sharma and S. K. Shivandu, "Integrating artificial intelligence and Internet of Things (IoT) for enhanced crop monitoring and management in precision agriculture," *Sensors International*, vol. 5, p. 100292, Jan. 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666351124000147

[6] H. M. Jawad, R. Nordin, S. K. Gharghan, A. M. Jawad, and M. Ismail, "Energy-Efficient Wireless Sensor Networks for Precision Agriculture: A Review," *Sensors*, vol. 17, no. 8, p. 1781, Aug. 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/8/1781

[7] A. Villa-Henriksen, G. T. C. Edwards, L. A. Pesonen, O. Green, and C. A. G. Sørensen, "Internet of Things in arable farming: Implementation, applications, challenges and potential," *Biosystems Engineering*, vol. 191, pp. 60–84, Mar. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511020300039

[8] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 551–591, 2013. [Online]. Available: https://ieeexplore.ieee.org/document/6248647

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Jan. 2000, pp. 10 pp. vol.2–. [Online]. Available: https://ieeexplore.ieee.org/document/926982

[10] ——, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, Oct. 2002. [Online]. Available: https://ieeexplore.ieee.org/document/1045297

[11] S. Lindsey and C. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *Proceedings, IEEE Aerospace Conference*, vol. 3, Mar. 2002, pp. 3–3. [Online]. Available: https://ieeexplore.ieee.org/document/1035242

[12] S. Mandal, A. Yadav, F. A. Panme, K. M. Devi, and S. Kumar S.M., "Adaption of smart applications in agriculture to enhance production," *Smart Agricultural Technology*, vol. 7, p. 100431, Mar. 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772375524000364

[13] J. Lloret, I. Bosch, S. Sendra, and A. Serrano, "A Wireless Sensor Network for Vineyard Monitoring That Uses Image Processing," *Sensors (Basel, Switzerland)*, vol. 11, pp. 6165–96, Dec. 2011.

[14] R. La Rosa, P. Livreri, C. Trigona, L. Di Donato, and G. Sorbello, "Strategies and Techniques for Powering Wireless Sensor Nodes through Energy Harvesting and Wireless Power Transfer," *Sensors (Basel, Switzerland)*, vol. 19, no. 12, p. 2660, Jun. 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6630942/

[15] I. Surenther, K. P. Sridhar, and M. Kingston Roberts, "Maximizing energy efficiency in wireless sensor networks for data transmission: A Deep Learning-Based Grouping Model approach," *Alexandria Engineering Journal*, vol. 83, pp. 53–65, Nov. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110016823009018

[16] A. Tanenbaum, N. Feamster, D. Wetherall, A. Tanenbaum, N. Feamster, D. Wetherall, A. Tanenbaum, N. Feamster, and D. Wetherall, *Computer Networks, Global Edition*. Harlow, UNITED KINGDOM: Pearson Education, Limited, 2021. [Online]. Available: http://ebookcentral.proquest.com/lib/sun/detail.action?docID=6432022

[17] S. Farahani, "Chapter 3 - ZigBee and IEEE 802.15.4 Protocol Layers," in *ZigBee Wireless Networks and Transceivers*, S. Farahani, Ed. Burlington: Newnes, Jan. 2008, pp. 33–135. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780750683937000030

[18] A. Naveed, “lora/part7.pdf,” Nov. 2022. [Online]. Available: https://www.slideshare.net/slideshow/lorapart7pdf/254464286

[19] F. P. Correia, M. S. d. Alencar, W. T. A. Lopes, M. S. d. Assis, and B. G. Leal, “Propagation Analysis for Wireless Sensor Networks Applied to Viticulture,” *International Journal of Antennas and Propagation*, vol. 2017, p. 1, 2017. [Online]. Available: https://www.academia.edu/57380089/Propagation_Analysis_for_Wireless_Sensor_Networks_Applied_to_Viticulture

[20] O. Katircioğlu, H. Isel, O. Ceylan, F. Taraktas, and H. B. Yagci, “Comparing ray tracing, free space path loss and logarithmic distance path loss models in success of indoor localization with RSSI,” in *2011 19thTelecommunications Forum (FOR) Proceedings of Papers*, Nov. 2011, pp. 313–316. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6143552

[21] G. Hiertz, T. Denteneer, L. Stibor, Y. Zang, X. Costa-Pérez, and B. Walke, “The IEEE 802.11 universe,” *Communications Magazine, IEEE*, vol. 48, pp. 62–70, Feb. 2010.

[22] N. Abramson, “THE ALOHA SYSTEM: another alternative for computer communications,” in *Proceedings of the November 17-19, 1970, fall joint computer conference*, ser. AFIPS ’70 (Fall). New York, NY, USA: Association for Computing Machinery, Nov. 1970, pp. 281–285. [Online]. Available: https://dl.acm.org/doi/10.1145/1478462.1478502

[23] team simpy, “simpy · GitLab,” Oct. 2024. [Online]. Available: https://gitlab.com/team-simpy/simpy

[24] C. Jaikaeo, “wsnsimpy · GitLab,” Dec. 2018. [Online]. Available: https://gitlab.com/cjaikaeo/wsnsimpy

Appendix A

Project Planning Schedule

Week	Date	Description
1	5/08 - 11/08	Define project and outcomes
2	12/08 - 18/08	Investigate material, resources, order hardware
3	19/08 - 25/08	Commence simulation programming, learn PCB design
4	26/08 - 01/09	Programming, prototyping, PCB Design
5	02/09 - 08/09	PCB Design, prototyping, start report
6	09/09 - 15/09	Literature review, program hardware
7	16/09 - 22/09	Printing PCB, program metrics for system
8	23/09 - 29/09	Assemble full system, report writing
9	30/09 - 06/10	Report writing
10	07/10 - 13/10	Report writing
11	14/10 - 20/10	Gather results
12	21/10 - 27/10	Report writing
13	28/10 - 03/11	Gather feedback and make adjustments
14	04/11	Report Submission

Table A.1: Project Planning Schedule

Appendix B

Outcomes Compliance

ELO Description	How it was achieved
1. Problem Solving	The project involves designing a routing algorithm that balances power efficiency with communication reliability. Problem-solving will be demonstrated by optimizing the network to handle node failures and variable environmental conditions while ensuring continuous data transmission.
2. Application of Scientific and Engineering Knowledge	The project applies principles of wireless communication, embedded systems, and energy-efficient computing. It will integrate knowledge of sensor networks, low-power communication protocols, and environmental monitoring to develop a robust system.
3. Engineering Design	The engineering design focuses on developing a scalable and adaptable sensor network architecture. The design process will include selecting appropriate hardware, creating an energy-aware routing protocol, and implementing a protocol for traffic management.
4. Investigations, Experiments and Data Analysis	Investigations include simulations to test the network's performance under different scenarios. Experiments will be conducted in a vineyard environment, followed by data analysis to evaluate the effectiveness of the implemented protocol.
5. Engineering Methods, Skills and Tools, Including Information Technology	The project used engineering methods and tools to design, simulate, and implement the sensor network. Key skills in areas such as embedded systems, wireless communication, and power management are applied throughout the project. Computer packages such as SimPy was used for modelling, simulation, and information handling, while Python was used for computation and Matlab for data analysis. Additional tools like Kitty and VSCode were used for remote terminal server connection to the nodes.
6. Professional and Technical Communication	The project consists of a written report. Additionally, an oral presentation is included. This shows the ability to demonstrate competence and communicate effectively, both orally.
7. Individual Work	Primary responsibility was taken for successful completion of all aspects of the project.
8. Independent Learning Ability	Understanding information about signal propagation, the Fresnel Zone, and various routing protocols. Using this knowledge to design a solution to this problem.

Table B.1: ECSA Exit Level Outcomes (ELOs)