Bachelor Degree Project

Designing a Novel RPL Objective Function & Testing RPL Objective Functions Performance.

Author: Khalil Mardini & Emad Abdulsamad
Supervisor: Arslan Musaddiq & Fredrik Alhgren
Semester: VT 2023
Subject: Computer Science

Abstract

The use of Internet of Things systems has increased to meet the need for smart systems in various fields, such as smart homes, intelligent industries, medical systems, agriculture, and the military. IoT networks are expanding daily to include hundreds and thousands of IoT devices, which transmit information through other linked devices to reach the network sink or gateway. The information follows different routes to the network sink. Finding an ideal routing solution is a big challenge due to several factors, such as power, computation, storage, and memory limitation for IoT devices. In 2011, A new standardized routing protocol for low-power and lossy networks was released by the Internet Engineering task force (IETF).. The IETF adopted a distance vector routing algorithm for the RPL protocol. RPL protocol utilizes the objective functions (OFs) to select the path depending on diffident metrics. These OFs with different metrics must be evaluated and tested to develop the best routing solution. This project aims to test the performance of standardized RPL objective functions in a simulation environment. Afterwards, a new objective function with a new metric will be implemented and tested in the same environment conditions. The performance results of the standard objective functions and the newly implemented objective function will be analyzed and compared to evaluate whether the standard objective functions or the new objective function is better as a routing solution for the IoT devices network.

Keywords: :Routing protocol for low-power and lossy networks (RPL), Wireless Sensor Networks (WSN), The Internet of Things (IoT), Objective Function (OF), Low-Power and Lossy Networks (LLNs), Expected Transmission Count (ETX), Minimum Rank with Hysteresis Objective Function (MRHOF), Packet Delivery Ratio (PDR), Directed Acyclic Graph (DODAG)

Preface

We want to take the opportunity and thank our supervisors, Arslan Mussadiq and Fredrik Ahlgren from Linnaeus University, for continuously helping us in enhancing our research project and providing us with beneficial feedback. We also express our gratitude to Daniel Toll, who has guided us on the right path by providing useful suggestions and inspiration. May this research be valuable among pupils and scholars interested in the IoT field. :)

Contents

1 Introduction

The use of Internet of Things systems has increased to meet the need for smart systems in various fields, such as smart homes, intelligent industries, medical systems, agriculture, and the military. IoT networks are expanding daily to include hundreds and thousands of IoT devices, which transmit information through other linked devices to reach the network sink or gateway. The information follows different routes to the Internet of Things (IoT) which is a significant segment of Information Technology in our decade. It aims to transform real-world devices into smart virtual objects, unifying almost everything in the world under one infrastructure, and it keeps us updated about the current state of the devices plus to control the main functionality of the devices remotely [1].
Recently, the IoT has been used extensively in various real-life domains, especially in crucial environments such as medical departments, smart homes, and military departments. IoT devices with limited power resources, processing, and memory capacity work as capture points and transfer captured data without applying any processing or filtering to the system management server. The IoT devices are linked together to build a network that transmits information, known as low-power and lossy networks (LLNs). This network consists of a vast number of IoT devices that generate and transmit huge amounts of data that drain various network resources, especially energy, and power. LLNs need a superior routing solution that preserves IoT network resources. In 2011, Routing Protocol for Low-Power and Lossy Networks (RPL) is a standardized protocol developed by the Internet Engineering Task Force (IEFT). [2]. Since the routing is known to be as the pillars of the IoT network structure and is responsible for routing decisions, a wrong decision will lead to a waste of IoT network resources. Thus, researchers are interested in studying and developing the RPL protocol, which has been adopted as the actual routing protocol for IoT networks. Consequently, this protocol still needs further improvements in terms of routing path selection and power consumption. Statistics show a rapid increment in the number of IoT devices worldwide every year, as shown in Figure 1.1, which makes finding the best routing solution a vital task.

This paper is a 15 HEC bachelor thesis in computer science that mainly focuses on the RPL protocol objective functions performance in a simulation environment with different matrices to evaluate the best of them, which will be critical in the world of IoT.

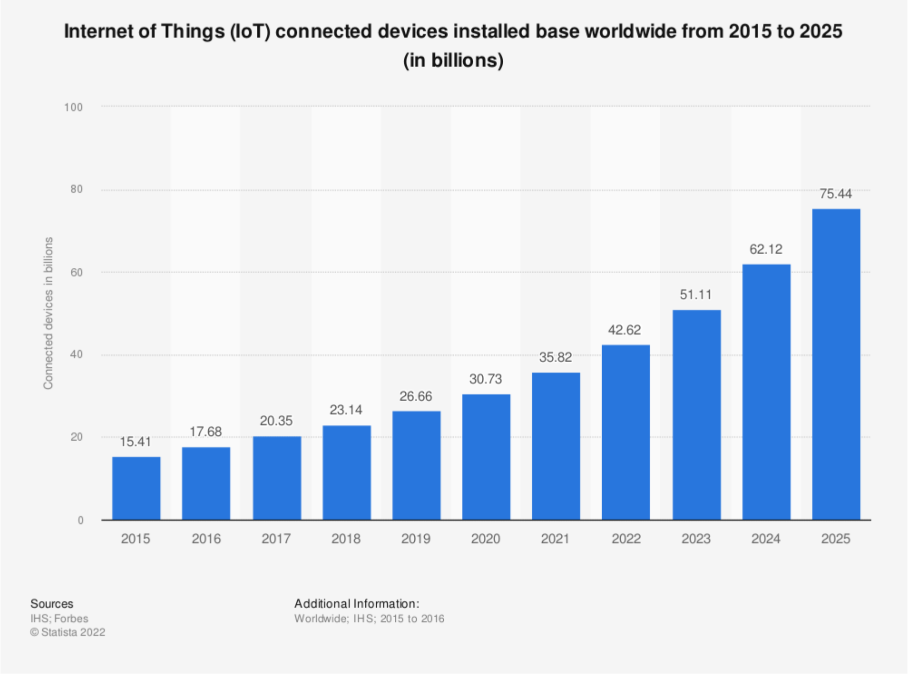


Figure 1.1: Number of installed IoT devices worldwide [2].

1.1 Background

The wide spread of The IoT raises attention to its ability to collect and exchange data throughout the globe; it also enables various devices and gadgets to be monitored and controlled remotely through the internet. The domain of IoT is used in broad areas in our society, such as agriculture, healthcare, environmental monitoring, and the industrial field. Monitoring trees and plants in big farms is automated using IoT sensors. Each tree has a sensor to measure humidity, temperature, and light intensity; moreover, IoT sensors are used for precipitation rate and alarm for incidents such as fires in forests. Hospitals must constantly monitor each patient's heartbeat rate and blood oxidation rate. In the industrial field, IoT sensors are used to control and monitor various machines in factories. IoT sensors in farms, forests, and factories are installed over long distances. On the other hand, hospital patients need to move between different sections and rooms; therefore, cord or wired connection cannot support long-distance connection or mobility [1]. The IoT mainly relies on wireless communication technologies especially designed for LLNs -such as Zigbee, LoraWan, Bluetooth-low-Energy (BLE), and IEEE 802.15.4- to establish connections among other IoT devices. Data transmission from IoT sensors to the system's main server or gateway needs network infrastructure to carry the transmission. The IoT sensors will connect to each other to build LLNs which consist of all IoT nodes. One of these nodes is at least connected directly to the IoT system management server or gateway, known as the IoT network root node [3].

LLNs utilize multi-hops to establish the transmission from nodes to the root nodes;. However, minimum multi-hops are widely used as a metric in many routing protocols, such as RPL RIP1 and RIP2. Routing information protocol (RIP) is a computer network routing protocol [4]. In a multi-hops connection, each node will build a parent list of devices that can be a part of a multi-hops path or route and classify them according to their rank and the number of hops the sensor needs to reach the root node. The multi-hop connection must be managed by rules and instructions represented as routing protocol [2]. Among the most popular and fast-growing routing protocols devoted to IoT is the RPL protocol due to its inherent ability to handle networks within constrained nodes. RPL protocol contains objective functions to manage and determine the best path to transmit the data. The first function used is Objective Function Zero (OF0), which uses the minimum hop count as a metric. The second function used is Minimum Rank with Hysteresis Objective Function (MRHOF) which utilizes the Expected Transmission Count (ETX) as a metric [5].

With our broad knowledge of computer science and average experience in the IoT domain, entering the field of IoT through building an LLNs network and testing the performance of connection protocol is a significant challenge. Moreover, coming up with a new RPL objective function metric suitable for low-power IoT networks in different terms, such as energy consumption, latency, and packet delivery, is another challenge we might face during the testing process.

1.2 Related work

Muneer et al [5]. Stated that the performance RPL object functions that include OF0 and MRHOF were evaluated in terms of Packet Delivery Ratio (PDR) and power consumption and were conducted in two different network topologies: the grid and random. The Packet Recipient ratios (RX) were implemented on 50 and 65 nodes. So, in their study, they focused on the impact of various parameters such as Packet Reception Ratio (RX), network topology, object function and the number of nodes that has an area of 500x500 meter square area plus 100 meters for interference range and 75 meters of transmission range (TX). Their experiment results displayed that OF0 exceed MRHOF with respect to power consumption meanwhile keeping roughly the exact same performance of PDR.

P. Janani et al [6].Decided to use different metrics, hop count and link quality to select the transmission path to send data. They worked on testing the environment, which includes 20 IoT nodes distributed over and 150x150 area. The IoT nodes were running on Contiki as an operating system. The nodes select different paths and build the Destination Oriented Directed Acyclic Graph (DODAG), which shows a tree structure graph. The experiment is done by increasing the traffic between the source nodes and root node and has been repeated with a different number of nodes. While testing was in progress, a few parameters were monitored:

- Average throughput
- End to end latency
- Packet drop
- Loss ratio
- Energy consumption

The experiment shows that the end-to-end latency for OF0 with the hop-count metric is higher than the latency for OF with the link quality metric. The packet drops, and loss ratio for OF0 with the hop-count metric is also higher than that for OF with the link quality metric. While the energy consumption rate was the same for both OFs with the hop-count metric and link quality metric.

Qasem et al [7]. They investigated the performance of RPL in terms of the two main OFs (MRHOF and OF0) under two random and organized network topologies within a light network density. Numerous OFs parameters are used in this experiment: packet delivery ratio (PDR), Packet reception ratio (RX) and Power consumption. The evaluation of this experiment was performed based on the network topology and was compared for the two OFs. The simulation results showed that light network density does not generate enough data transmission, which leads to different RPL performance when OF0 and MRHOF are applied in organized and random topology networks. The power consumption for OF0 reached 1.23% compared to the MRHOF, which reached 1.15%, while the PDR for the OF0 is 0.99 compared to MRHOF which is 0.98. The result shows that RPL performance in a light density network in utilizing the MRHOF can offer ideal performance compared to OF0 in terms of power consumption and PDR. So, the experiment shows that RPL performed equally within the two main OFs in a light network density ranging between 30 to 40 nodes for the RX 60% when applying Grid and Random topology. Moreover, The RPL performance tuned out to be efficient in relation to power consumption and PDR in using 60% RX.

Jara et al [8]. Studied and analyzed the RPL performance based on the two standardized objective functions OF0 and MRHOF under various scenarios, such as node positioning and mobility, in terms of various OFs performance metrics which are packet loss, power consumption, Expected Transmission Count (ETX) and Traffic overhead. The analyses of the experiment showed that distributed nodes using MRHOF in liner position use more energy consumption than manual, ellipse, and random positioning. In contrast, nodes using the OF0 use the same amount of energy consumption in all different positions. Also, the MRHOF transfer more data traffic in linear positioning than in ellipse, random and manual positioning. As opposed to the MRHOF, OF0 send less traffics in all positions. So, the experiment results display that MRHOF performs better within scalable node positioning networks, while the OF0 performs better in mobile node networks. Moreover, for all metrics, both objective functions produce approximate results, with the only difference being of linear position, which utilizes extra energy and has a high control traffic overhead. Due to the reason that an objective function cannot satisfy the RPL protocol requirement it raises the need to produce new OF that support more mobility of the nodes and enhances the RPL protocol in terms of applications.

1.3 Problem formulation

The widespread usage of RPL routing protocol to build IoT networks raises the importance of testing the performance of used RPL OFs that elect paths and transmit the data over the IoT network. Path selection plays a primary role in the transmission properties such as transmission latency, loss ratio and energy consumption. Due to the RPL OFs method lacking comprehensive testing, the RPL OFs performance needs to be tested, and new metrics need to be carried out to improve the RPL protocol. Consequently, this will support Computer Science (CS) workers, especially those working in the IoT domain, and help build the best IoT communication infrastructure and future research. Most of the related work of RPL OFs focused on evaluating the OF's performance without trying to conduct a new OF´s metric, collect relevant parameters data for different OFs, and analyze them to assess the most proper OF metrics for multi-hop connections and contribute to better knowledge for future studies. So, this paper will try to answer and illustrate the following research objectives:

- Run RPL protocol and evaluate the performance of standardized OFs.
- Propose a new OF, evaluate the new OF, and compare the new OF's performance with standardized OFs.

1.4 Motivation

Due to the high usage of IoT in different sectors of society, such as Healthcare, smart applications, and the military, it mainly relies on a vast number of low-powered sensors that lead to improved lifestyles and present new services for the community[1]. The IoT is one of the Industry Revolution 4.0 factors, and most companies have started moving to smart factories. Smart factories will include systems with huge numbers of IoT sensors, hundreds or maybe thousands; routing protocol performance in these huge networks will hardly affect production and financial profits for each factory [9]. The RPL rapidly gained much interest, and numerous research were conducted to enhance and evaluate performance in various applications. Evaluation of a single function performance will not provide accurate results compared to other OF's performance. Improving the RPL protocol will be the base and guarantee for facilities, organizations, and factories to move ahead to smart IoT systems.

1.5 Results

This research paper illustrates the stages of testing and analyzing RPL OFs. Based on our predictions. This study starts by evaluating and estimating the performance of the standardized RPL OFs on the IoT LLN network. Afterwards, developing a new RPL OF and evaluate this OF to compare it with standarized OFs and check whether the new OF performs better than the standardized OFs. The RPL OF that achieves better results and performance will highly contribute to the development of various IoT domains. Research hypothesis in terms of RPL objective functions performance will be formulated and based on the final results of the experiment, the hypothesis will be accepted or rejected.

Results of this research are significantly based on designing and implementing a new OF and technique that could be an alternative to the pre-defined standardized RPL OFs. In this research paper, a new RPL OF will be proposed which could enhance the Computer Science (CS) area, specifically the IoT domain and network infrastructure.

1.6 Scope/Limitation

In the process phase, we started by preparing the work environment and other requirements to apply the RPL OF performance testing. As the first step, we set up our control system, endpoint, or server, which is Contiki-OS. Our WSN network will include 100 IoT nodes in a simulation environment and one node that will work as network root (sink node). The network nodes will be configured to send constant data with fixed packet sizes and a specific time interval. During the test, we will collect information about several performance factors of RPL OFs, such as packet delivery, latency, energy consumption, and packet drop.

This research paper will apply the test experiment on a certain number of virtual nodes, which will be done in a simulation environment, and the nodes will not send accurate sensor measurement data. Moreover, the test will only be applied to the RPL protocol and not compared with other routing protocols. The data transaction will rely on a local network during our project experiment.

This thesis project aimed to test RPL objective function performance on real IoT devices, Arduino nano BLE. Contiki-OS is selected as an operating system for the research experiment. Contiki-OS was initially released in 2003. During the last twenty years, the developer, Adam Dunkels, did not work on developing, updating, and testing the operating system or supporting upcoming new IoT devices. We did not manage to install and run Contiki-OS on the Arduino nano BLE device. Then we contacted our supervisor Arslan Musaddiq regarding this issue, who replied that the IoT device vendor does not support Contiki-OS. Due to this issue, we had a meeting with the supervisor to discuss our alternative options, and we finalized the meeting on applying our experiment in a simulation environment.

1.7 Target group

The leading target group in this thesis paper is the people in the computer science field, especially those who work in the IoT domain and technology industry that primarily produce smart wireless network sensors and home automation devices, which all fall under the definition of IoT. This thesis paper evaluates various approaches regarding RPL OFs performance and could be used for future research.

1.8 Outline

The remaining part of the thesis paper will be organised as follows: Chapter 2 will cover the research Methodology part and present what methods are picked and how they are applied in the research paper. Moreover, the validity and reliability will be covered. Chapter 3 will outline the theoretical background of the main principles related to the research of the state-of-practice and state-of research. Also, identifying and explaining the relevant knowledge gap related to testing RPL objective functions performance. Chapter 4 will explain the implementation of the experiments for the standardised objective functions and the new proposed objective function and the technics that helped in collecting data. Chapter 5 will contain two main sections, which present the performance test results; the first section presents the results of testing and running OF0 and MRHOF objective functions, and the other section will present the results of the new defined objective function. In Chapter 6, the results will be analysed for the two standardised objective functions and the new objective function then the results will be compared to evaluate the most appropriate metric. The purpose of Chapter 7 will be to answer the research objectives and examine which RPL OF performs best based on the results from Chapter 5. Finally, chapter 8 will offer a new efficient objective function that will improve the RPL OF performance.

2 Method

This chapter starts by selecting the appropriate research methodology along with illustrating and defining the selected research methodology approach that is applied in this study, then a more thorough description of the methodology procedure and how would it be carried out. This chapter concludes by discussing and motivating the validity and reliability.

2.1 Method Selection

The thesis topic of this paper is deeply specialized within the IoT field and communication domain. Since most of society does not have enough knowledge regarding this topic, the questionnaire survey method would not work in gaining proper evaluation from society. Additionally, to the shallow knowledge in society, an interview with a specialized researcher will reflect the researchers' point of view depending on their beliefs or personal experiments with unknown situations and conditions, which cannot provide an accurate evaluation of this research.

The Systematic Literature Review methodology compares previous research results done by other researchers, which will not add new results or improve the topic of this research.

The Case Study methodology will impose certain environments with un-controlled conditions and impacts that can lead to inaccurate results during the research process.

The Controlled experiment allows us to define the experiment environment and control various variables that can affect experiment results, allowing us to repeat the experiment under the same conditions and apply changes for a certain variable. Therefore, a control experiment is the most appropriate methodology for this research [10],[11].

2.2 Controlled Experiment

The controlled experiment is a scientific method to test one or several hypotheses in controlled environment conditions to study the effect of manipulating experiment-independent variables on the dependent variables and determine the relationship between these variables. A controlled experiment is repeatable and applicable with the same condition and variables to provide precise and accurate results. This methodology is widely used in various fields and sciences [12]. The controlled experiment methodology consists of several experimental process stages described further in this section.

2.2.1 Scoping

In the scoping phase, it is vital to define a clear hypothesis and set the goals of the experiment based on the research question. Scoping requires specifying variables that need to be measured or altered, along with the conditions under which the experiment will be carried out [12]. The objective of the scooping goal is to guarantee that the significant aspects of the experiment are specified prior to the two main experiment process phases, which are planning and execution. This project aims to develop a new objective function metric that will improve the performance of the OF in the RPL protocol and evaluate which OF metric performs better.

2.2.2 Experiment Planning

2.2.2.1 Hypothesis Formulation

The goal of the controlled experiment is to evaluate the effects of manipulating one or more different variables in the experiment functions. Researchers come up with one or several hypotheses through theoretical studies of improving the performance of a system by changing different variables [12]. The rapid development of the IoT systems may lead to several issues, and the standardized OFs become inefficient in selecting routes within the IoT network. The functionality of the IoT systems may get affected therefore, we have revised the RPL protocol and the OFs and came up with the following hypothesis:

General performance

- H0: The RPL standardized OFs has a better performance than the new proposed OF.
- H1: The new proposed OF has a better performance than the standardized objective functions OF0 and MRHOF.

Managing power consumption and average energy usage

- H0: The RPL standardized OFs has a better performance in terms of managing power consumption than the new proposed OF.
- H2: The new proposed OF is more efficient in managing IoT devices power consumption and average energy usage than the standardized OFs.

The current OFs are not that effective and may lead to the rapid development of the IoT to several issues that affect the functionality of the IoT system. A new objective function or metric may perform better regarding power consumption, latency and packet delivery.

2.2.2.2 Variables Selection

The researchers need to define all variables involved in the applied experiment and classify them. Variables are classified as the independent variables, which are controlled and manipulated by the researchers, and the dependent variables which are affected by the manipulation of independent variables [10]. In Our project, the independent variable is the ranking metric of the objective function, and the dependent variables are the performance properties such as delay, arrival packet ratio, latency, and power consumption.

2.2.2.3 Experiment Design

In the experiment design, the researchers refer to how the independent variables manipulation is done in the process to test the hypotheses of the research [12], [13]. In our experiment, we will use fixed variables and experiment environment factors and change the RPL objective function or the ranking metric and evaluate the performance's properties.

2.2.2.4 Validity Evaluation

The researchers need to control all variables and factors in the experiment and choose a trusted collect data method to confirm the result's validity [12].

2.2.3 Experiment Operation

The experiment operation consist of the following sub-sections as follows:

2.2.3.1 Preparation

The experiment and surrounding environment should be prepared thoughtfully, including all related variables and factors before applying the experiment [13]. The experiment will be conducted in a simulated environment that is isolated from external influences such as jamming and interference. The experiment operation consists of the following sub-sections as follows:

2.2.3.2 Execution

The experiment will run on a virtual fixed number of IoT devices and fix network topology several times for each of the standardized objective functions plus for the new objective function.

2.2.3.3 Data collection

Through the experiment, the researchers collect all data about performance properties for analysis and set the results [13]. This experiment will use a non-GUI simulation environment, which will record all the network events and logs in log files during running the experiment.

2.2.4 Interpretations of Results

The collected data will be checked and compared for validity and then analyzed to draw conclusions and extrapolate the experiment's results [12]. The experiment results are used to evaluate the hypotheses of this research.

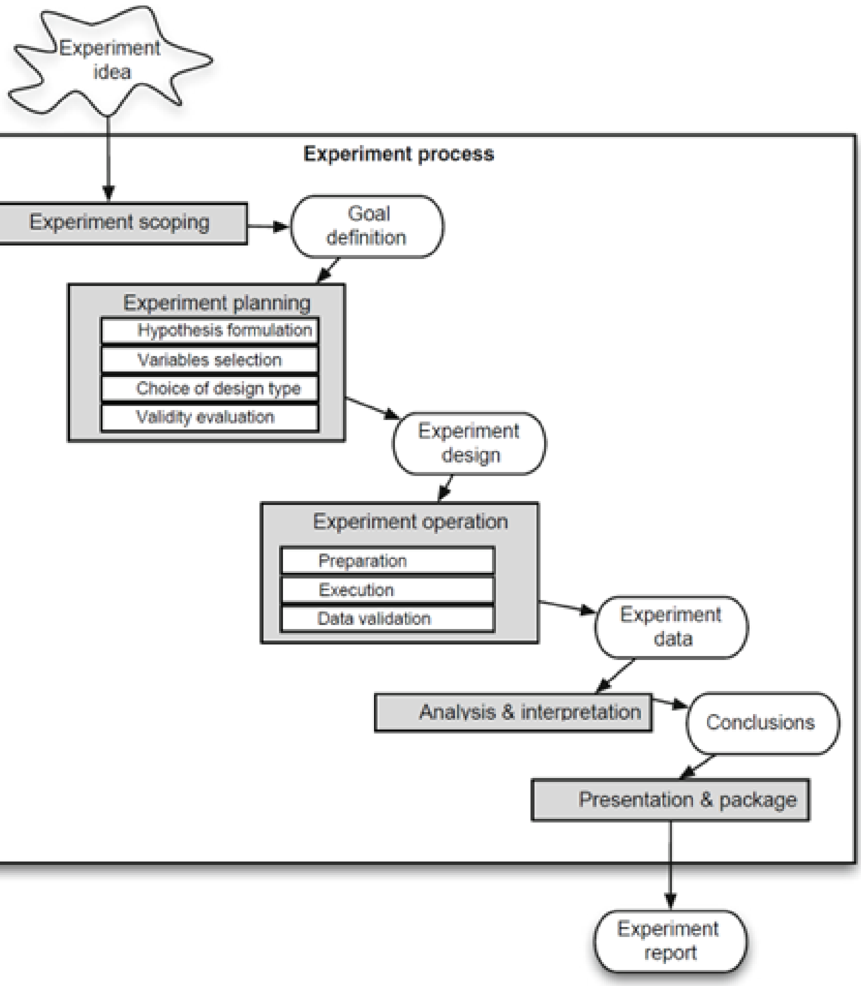


Figure 2.2: Overview of the experiment process methodology

Figure 3.5 illustrates the experiment process stages conducted in this research; first and foremost is to set the experiment idea by defining the experiment perspective, domain, scoop, and purpose. This experiment's main goal is to develop a new objective function metric and evaluate it with the standardized objective function. Later is to set experiment planning by defining the research's hypotheses, selecting the experiment variables, and selecting the most influential research experiment design type. Secondly, conduct the research experiment by applying data collection and preparing a suitable experiment environment. Also, to guarantee an efficient execution process and avoid faulty data input or implementation difficulties during the experiment operation. The final stage of the experiment is to analyze the collected data and present the final result as an experiment report specifying the entire process.

2.3 Reliability

Reliability indicates the extent to which research results, findings and conclusions can be trustworthy and replicated. Developers and researchers depend mainly on research and experiment results to evaluate the different routing protocols in the LLNs, so reliability is a critical aspect when applying an experiment which contributes to accurate and genuine results by including accuracy to the simulation model and quality of the simulation input data within the experiment. To achieve the experiment's reliability the researcher and tester need to check the experiment environment with all the surrounding conditions that may effect on the experiment. In the second step, the researcher designs the experiment scenarios while considering all influencing factors. As the third step, the researcher will verify the input data and confirm that it is applicable in the real world and can be replicated; in addition, by checking the accuracy and validity of the data collection methodology. In this research study, we tested the Contiki-OS on different platforms and tried to apply simple tasks to validate the stability, efficiency, maintainability, and performance. In the real world, each sensor of the LLN network will be placed according to the needs in a certain place, so the distribution of the sensors will be organized. Consequently, The LLN network will be built in an organised distribution to avoid network issues genrated by bad IoT sensor distribution such as Bottleneck or sensors out of coverage. We will design experiment scenarios that is applicable in the real world, such as factories and hospitals, and compare the result data to other experiment results by other researchers. Not following the research experiment steps carefully will violate the experiment results, such as choosing the wrong method of collecting data, selecting the wrong input data, or choosing a different experimental environment than what the research required.

2.4 Validity

In this experiment, several internal and external factors might affect the validity. Other systems in the surrounding environment can generate noise and interference that badly affect the experiment results, which does not exist in our experiment due to using a simulation environment. We will conduct our experiment frequently in the same system and platform which lead to avoid different conditions and factors in various system environment. Automated data collection will guarantee accuracy in gaining results. The perfect performance of the data transmission can be easily determined by calculating the number of transmitted packets in a time as expected results. The experiment results can be compared with the perfect performance results to ensure the validity of these results.

The RPL protocol was tested and standardized by Internet Engineering Task Force (IETF) in March 2012. This means the validity of the RPL protocol functionality is guaranteed [14].

The operating system used for IoT devices is Contiki, which was implemented and initially released in 2003 and widely utilized in the field of IoT; it also improved the validity, reliability, and performance of data transmission. Contiki is an open-source operating system dedicated to operate on IoT devices with limited capabilities in low-power and lossy networks (LLNs). Contiki supports simulation environment and became widely accredited by the research community [15].

3 Theoretical Background

This section will present a brief theoretical background within the examined area of testing RPL protocol for different objective functions performed on an IoT network. So, this section will illustrate the critical concepts related to communication technologies, routing protocols, RPL protocol, DODAG topology, and objective functions principles. Furthermore, these principles will be defined and analyzed so then it will provide a comprehensive image for the reader:

3.1 IoT Network

IoT technology allows heterogeneous devices to connect to the internet and between each other. IoT systems have become increasingly integrated into various areas, such as alarm systems, intelligent medical sensors, and smart home sensors. Also, the number of connected IoT devices increases swiftly every year and is expected to reach 40 billion devices by 2025 [14]. IoT's mobility and agility advantages make integrating within any system desirable and convenient. IoT systems have become the backbone of the fourth industrial revolution with new automation, control, and monitoring features through the global network [16]. IoT devices link together to exchange and transmit data with each other, system controller, central server, or cloud-based platform. The connected IoT devices build the necessary infrastructure to utilize single or multi gateways to transmit data to the control center known as Low Power and Lossy Networks (LLNs).

3.2 Low power and Lossy Networks (LLNs):

LLNs are several interconnected IoT devices, such as sensors and controllers, with constraints in power supply, processing capability, storage capacity, and memory size. The IoT devices capture and collect data and transmit this data to the network gateway, a similar IoT device called a root or an LLN border router (LBR). The general communication pattern in the network is multi-points-to-point (MP2P) [17]. IoT devices support low-power transmission and use different communication technologies, such as Zigbee, Bluetooth Low Energy, LoRaWan, SigFox, and IEEE 802.15.4e 1mW as maximum transmission power. As a result, LLN characterizes communication with short range, low data rate, limited packet size, asymmetric links capability, and high packet loss [18].

3.3 IoT communication technology

3.3.1 Bluetooth Low Energy (BLE)

The Special interest Group (SIG) was the first to develop and emerge one of the most critical wireless technologies that are Bluetooth Low Energy (BLE) [19]. The BLE is a wireless low-power technology established for low-range communication and monitoring applications; The BLE significantly relies on a single-hop communication method to route the data to its designated destination. The IETF 6LoWPAN Working Group (WG) has already acknowledged the significance of the BLE in the Internet of Things as the 6LoWPAN WG is creating a specification for transmitting the IPv6 packets over BLE [19].

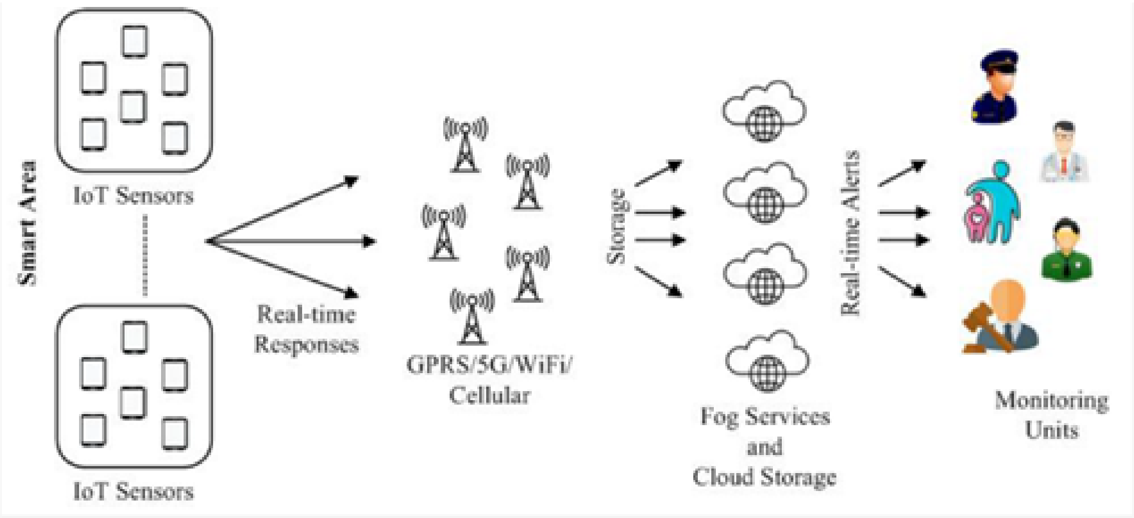


Figure 3.3: LLNs IoT network structure scenario [14]

3.3.2 ZigBee

ZigBee is a short-range wireless connectivity standard with low power consumption. The standard defines different carrier frequencies for ZigBee 868 MHz in Europe and 902-928 MHz in North America [20]. ZigBee supports network topologies, mesh, star, and cluster trees. Mesh topology leads to support multi-hop paths, and routes, which allow to build massive area networks with a vast number of nodes, up to 65000 nodes, and provide a connection for long-distance nodes by multi-hops [20]. ZigBee has a low data transfer rate of 20 – 250 kbps which fits the WSN networks, where sensors send a small amount of measurement data each period. Zigbee allows nodes to enter idle mode using energy detection (ED) and link quality indication (LQI) to re-activate the radio transceiver when required.

3.3.3 LoRaWan

LoRa is a low-power and long-range connectivity standard LPWAN. LoRa provides long-distance connections up to 20 KM. LoRa supports three bandwidths, 125 KHz, 250 KHz, and 500 KHz, and a data transfer rate from 300 bps to 37.5 kbps [21]. The coverage area of a LoRa node is extensive and may contain a large number of other nodes which may use the same bandwidth and channel, which may generate confusion. So, the active period of the radio transceiver is strict in the period. The implementation involves using a LoRa gateway to receive data transmitted by a sender node and then forward it to the server, as the sender radio transceiver cannot receive data from another sender [22].

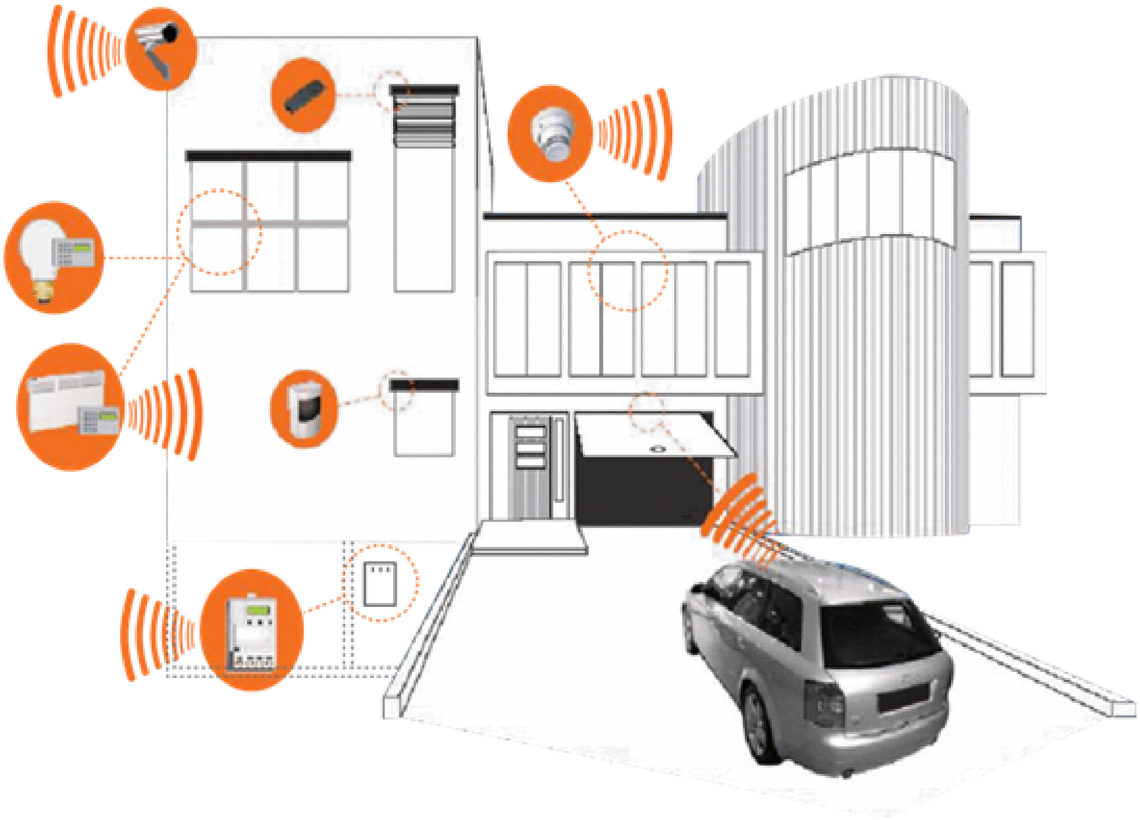


Figure 3.4: Smart home sensors and controllers se ZigBee, short range communication technology, to connect to the Smart Home management system [20].

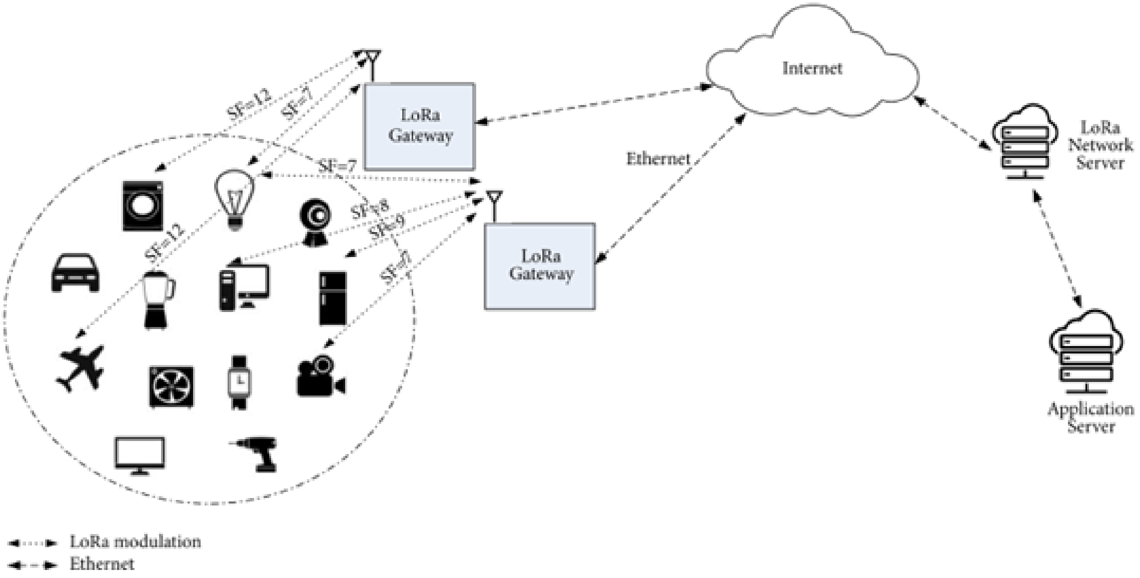


Figure 3.5: LoRaWAN Communication system and long distance IoT devices send data using LoRaWAN to the LoRa gateway and the IoT management system[22].

3.4 LLNs Routing protocols

Communication technologies provide a point-to-point physical connection between IoT nodes; The data packet generated in an IoT node in the LLN transmits through several nodes with a number of hops, and this transmission needs some directing to follow the right path to the destination. Directing or routing is the role of a routing protocol, which selects the best path in the network to transmit data packets through. LLNs are different than computer networks in terms of packet size, data volume, connection type (connectionless or connection-oriented), and hardware resources, so computer routing protocols do not fulfill the LLNs network requirements [23]. Standardized and non-standardized routing protocols for LLNs were created and developed to handle routing tasks such as CARP, IEEE 802.15.4e, RPL, and CoRPL [24].

3.4.1 Channel-aware routing protocol (CARP)

CARP is a routing protocol designed for underwater multi-hop wireless sensor networks used to select the best path for data transmission. CARP uses simple network information to choose the path, such as hop count; CARP also uses link quality between two nodes to select the best link [25]. IoT network nodes forward a data packet to the next node without saving routing information for subsequent transmission. So, each packet IoT node needs to check the best link to use. CARP is not a standardized protocol; therefore, CARP is not used in other IoT applications [24].

3.4.2 IEEE 802.15.4e

IEEE and 802.15 task group worked to redesign the existing 802.15.4 protocol to fulfill embedded devices networks and LLNs requirements and came up with the new standardized protocol IEEE 802.15.4e MAC. The new design inspired ideas from other protocols, such as slotted access, shared and dedicated slots, multi-channel communication, and frequency hopping, as well as the latest techniques of MAC behavior modes that include the following functional enhancements [26]:

- Low Energy (LE)
- Enhanced Beacons (EB)
- Multipurpose Frame
- MAC Performance Metric
- Fast Association (FastA)

And defined the new following MAC behavior modes:

- Time Slotted Channel Hopping (TSCH)
- Deterministic and Synchronous Multi-channel Extension (DSME)
- Low Latency Deterministic Network (LLDN).
- Asynchronous multi-channel adaptation (AMCA)
- Radio Frequency Identification Blink (BLINK)

3.5 Routing Protocol for Low-Power and Lossy Networks (RPL)

The RPL protocol was first created by the IETF group and based on IPv6; it is also a routing protocol that is designed for LLNs and wieldy used for several IoT applications due to its flexibility and ability to meet various network requirements [27]. The RPL allows the sender of the packet partially or wholly to determine the path that the packet follows through the network and allows the nodes to discover all the possible routes to the target node. RPL outlines a technique for creating a logical topology known as the DODAG network. Each DODAG represents a routing tree generated by the root node known as the network sink node. The RPL protocol uses different metrics to calculate the best paths, such as minimum hop count and Expected Transmission Count (ETX) [27].

3.5.1 DODAG Network

DODAG is the structure of an LLN network built by RPL protocol when the network IoT nodes are linked together. DODAG takes the shape of a tree starting from the root node. The building of DODAG is affected by the RPL protocol used OF which determines the best route to transfer data [28]. Each node will get a ranking number by the objective function which present the cost for the node to reach the DODAG root node. To build the DODAG network structure, IoT nodes exchange DODAG control messages to define the rank of each node and link to build the network. The RPL protocol allows upstream routing from the child node to the parent node and downstream routing from the parent node to the child node. Upstream routes are created during the first process of sending the DIO messages, RPL needs the network information in the DIO message to inform the child node which is the best parent node, so when the child node wants to send a packet to the root node it simply sends the packet to the best parent node and then the parent node send the packet to the next best parent node until the packet reaches the DODAG root node[29],[30].

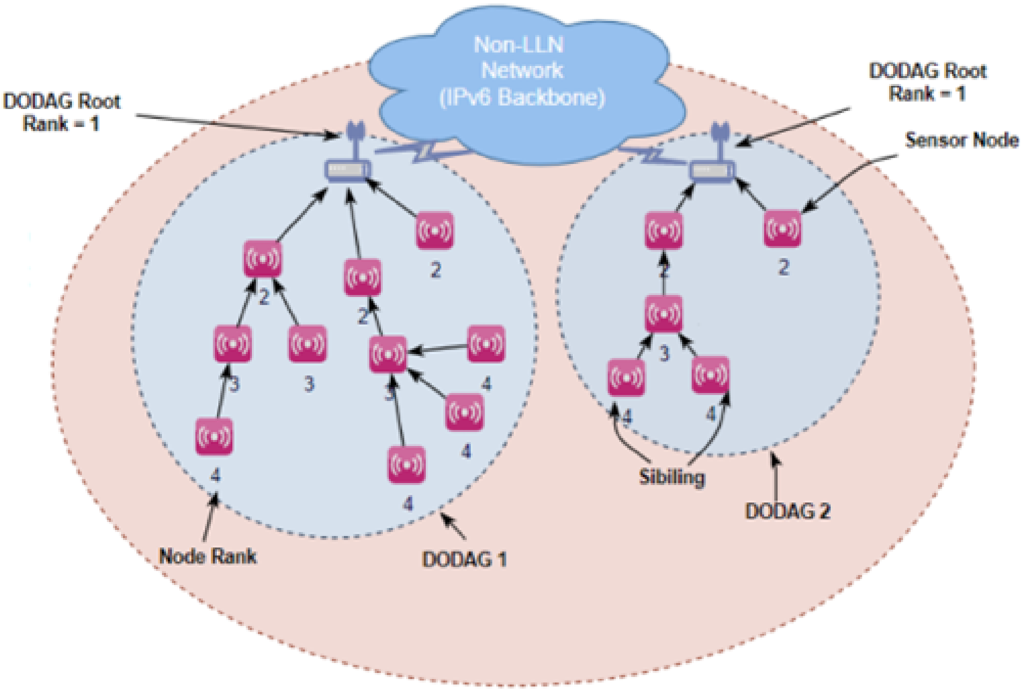


Figure 3.6: Two DODOG tree for two different LLNs with one root connected to the web for each LLN [31].

3.5.2 Building DODAG Network

Firstly, the root node constructs an DODAG network by sending DIO control messages to the neighboring nodes that contains data about node rank, each node that receives a DIO message has to process it and decide whether or not to join a DODAG according to the OF. In case the node joins the DODAG then it will have a path up to the root, and then the node will calculate its rank and update the neighbor node table and choose the best parent, which will be used to forward messages to the DODAG root. If a node is set as a router, it must update and resend DIO messages to its neighbors' nodes. Each node that receives DIO message and join the LLN network should send DIO message to the surrounding nodes and continue the same process until all nodes reach the DODAG network [32]. RPL allows nodes to join at any time. New nodes use a DIS message to request a DIO message from a linked node in DODAG network. Then the new node determines its best parent node by receiving a DIO message according to OF. The DODAG nodes send DIO messages frequently to maintain the stability of the DODAG network [32].

3.5.3 RPL Control Messages

The control messages are a new type of ICMPv6 information message with a Type of 155 that contain options used to join nodes and build the DODAG network such as protocol version, DODAG ID, and RPL type which define the type of the message [33].

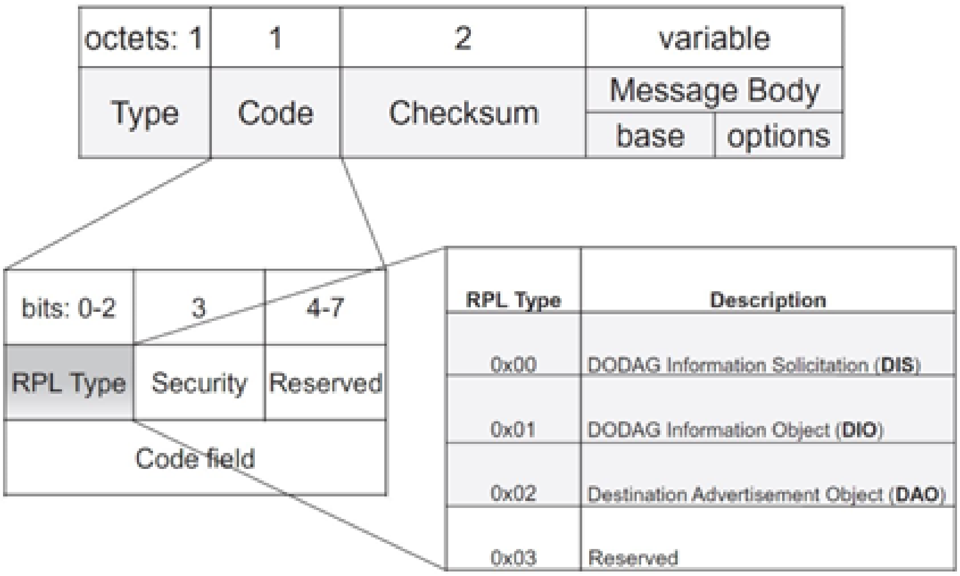


Figure 3.7: RPL control message structure.

3.5.3.1 DODAG information object (DIO)

The DIO message is mapped to the RPL message type 0x01. This message is used by root and parent nodes to build the DODAG structure and send the message downstream only. When the root node starts to build the DODAG or a node gets ranked, they send a DIO message to inform surrounding nodes about them. DIO message carries the node and network information such as node ID, root ID, DODAG version, node rank, DTSN flag, G flag, and mode of operation (MOP) which allow other recipient nodes to join the LLN network. The DIO is one-to-many messages[33],[34].

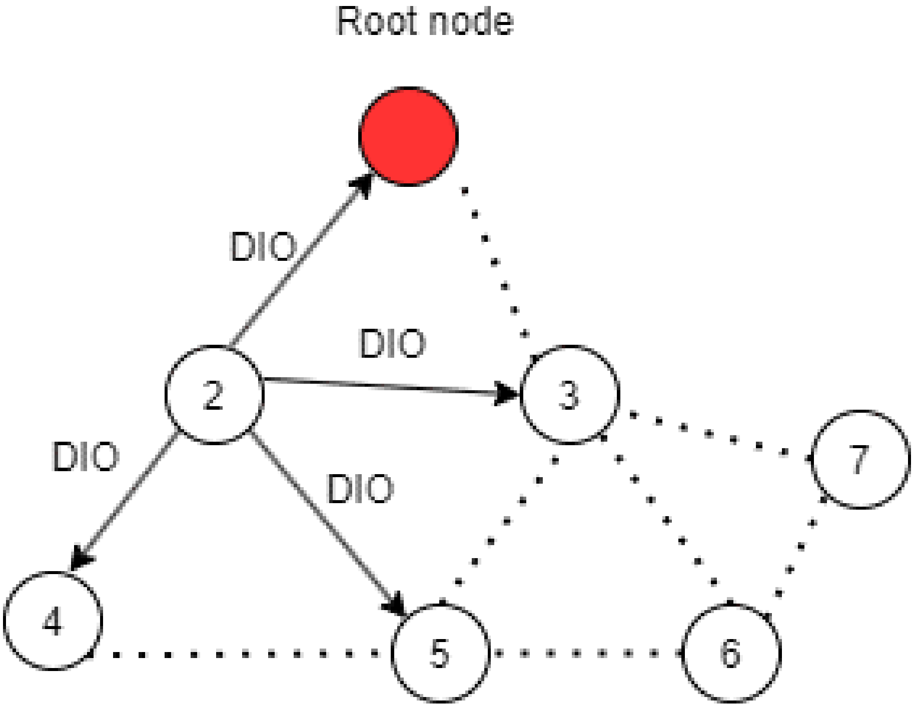


Figure 3.8: Node number 2 sending DIO messages for the neighboring nodes.

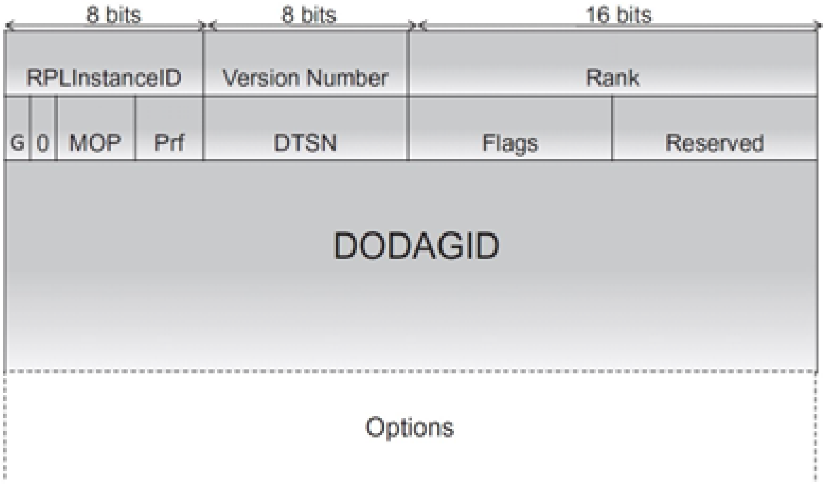


Figure 3.9: The DIO control message structure [33].

3.5.3.2 DODAG information solicitation (DIS)

The DIS message is mapped to RPL message type x00. DIS message is used by unlinked nodes or nodes that lost connection to ask for DIO from surrounding nodes. The DIS is one-to-many messages [33],[34].

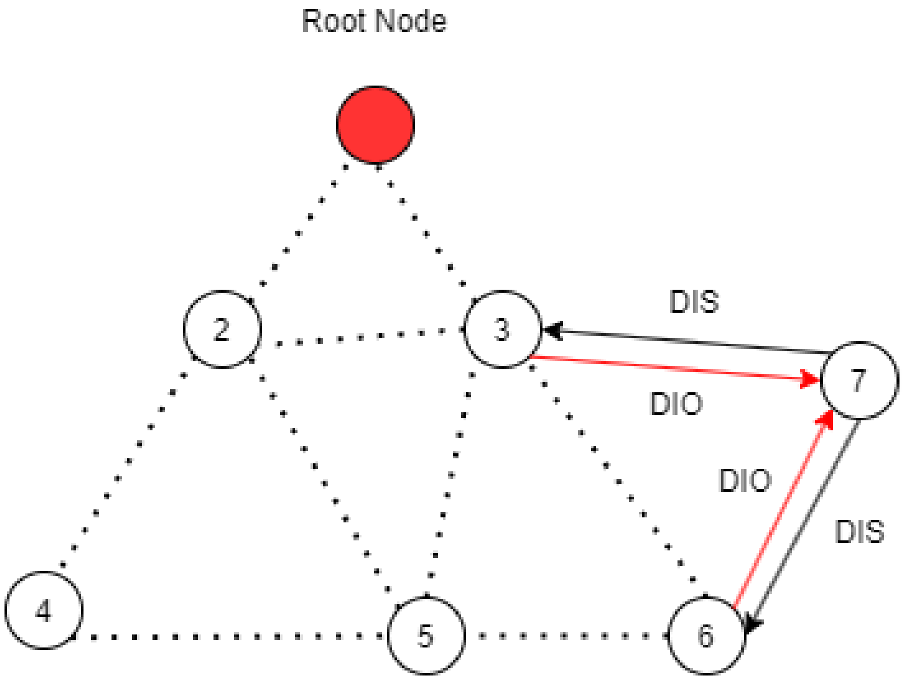


Figure 3.10: : Node number seven broadcasting to join the DODAG network by sending DIS to the neighboring nodes.

3.5.3.3 Destination Advertisement Object (DAO)

The DAO message is mapped to the RPL message type 0x02. The DAO message has two types DAO and DAO-ACK. The DAO message is used by the child node to inform parent nodes about the child existence and to be added to the child list. DAO is also used to check the connectivity between the parent node and the LLN root node. The DAO-ACK is sent as a reply or response from the parent or root node to the DAO message source node [33],[34].

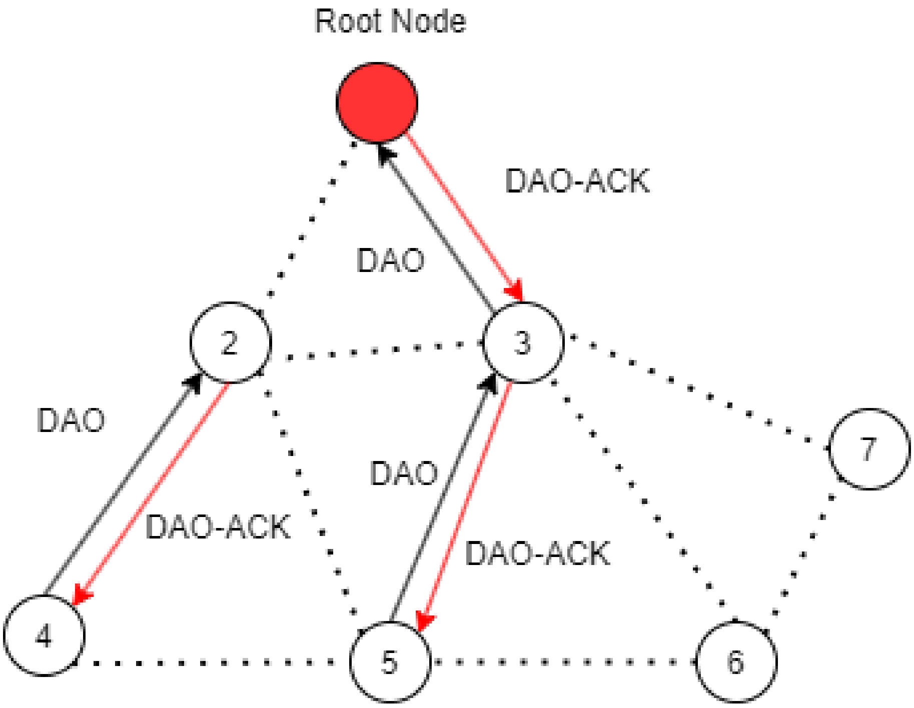


Figure 3.11: Illustrating the two types of DAO messages, where node number four tells the parent node number two about the existence of the child node, which is number four, afterwards the parent node number two send DAO-ACK message as a reply to node four.

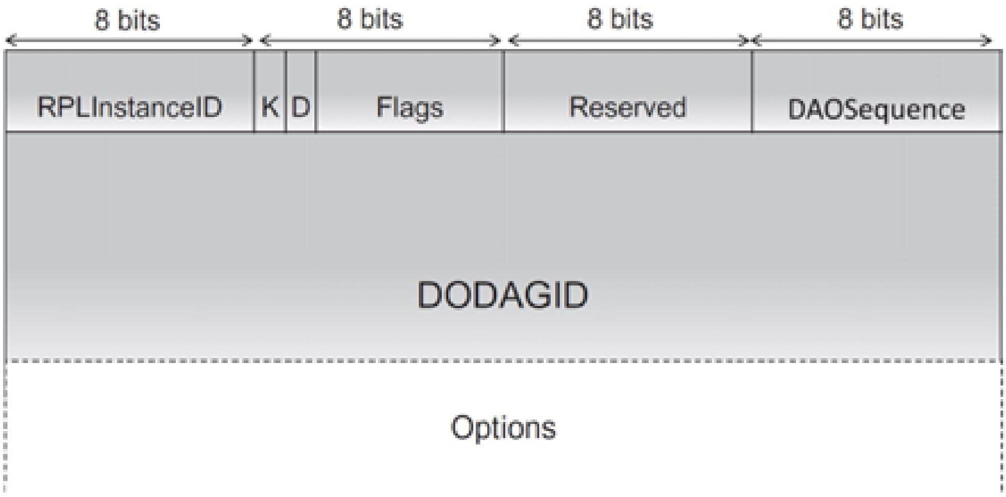


Figure 3.12: The DAO control message structure[33].

3.5.4 RPL Objective Function (OF)

RPL links IoT nodes in LLNs to build the DODAG and define routes the data is transmitted through. To select the best route or path in the network, RPL uses one function of a set of functions called objective functions. The objective function works to determine the DODAG and DODAG's version that the IoT device will join. Different objective functions use different criteria called metrics to rank the IoT nodes, which is the main factor used to create paths and avoid loops in the LLNs [28]. IoT nodes in the IoT network are classified as parent or child depending on the node rank, and then each node selects the preferred parent with the lowest rank. The paths in the IoT network are created according to the lowest cost of the path [35]. RPL protocol has two standardized OFs: Objective Function Zero (OF0) and Minimum Rank Hysteresis Objective Function (MRHOF).

3.5.4.1 Objective Function Zero (OF0)

OF0 is one of the first developed objective functions and uses the number of hops as a metric to determine the rank of the IoT nodes. The IoT node defines the preferred parents of neighboring nodes by selecting nodes with the lowest rank, and that creates the shortest path to the root node in terms of the number of hops for the node and builds a loop-free DODAG. OF0 use the following equation to calculate LLN nodes' rank [35].

$$R(N) = R(P) + rankIncrease$$

$$rankIncrease = (Rf * Sp + Sr) * MinHopRankIncrease$$

- R(N): Node Rank
- R(P): preferred parent Rank
- RF: Rank factor
- Sp: Step of rank
- Sr: Stretch of rank

OF0 may select paths with poor links quality in the LLN because IoT nodes choose the path with the minimum number of hops even if it is unreliable, thus more re-transmissions are produced, and more packets are lost.

3.5.4.2 Minimum Rank with Hysteresis Objective Function (MRHOF)

MRHOF is proposed by IETF to solve problems for fixed metric, objective functions such as minimum hop count. MRHOF uses minimum Expected Transmission Count (ETX) to rank IoT nodes and build the DODAG in the IoT network [36]. ETX metric expresses the required number of transmissions to deliver the data packet to the destination successfully. The IoT node selects the preferred parent with the lowest value for ETX. Each IoT node calculates the ETX for all possible parents using the following equation [37]:

$$ETX = \frac{1}{Df * Dr}$$

- Df: forward delivery ratio
- Dr: reverse delivery ratio

The link or path quality is measured by the ETX value, which is called the link cost.Moreover, having a low ETX cost suggests a more reliable link.

4 Research project – Implementation

The following Section will present the procedures of testing the RPL standardized objective function performance for OF0 and MRHOF, next is to design and implement the new proposed objective function metric.

4.1 Software and hardware:

The experiment process has been conducted on a computer with the following hardware specifications:

OS:Windows 10 Pro 64-bit
Processer: AMD Ryzen 7 2700X
RAM: 16 GB
Hard Drive: Crucial CT500MX500SSD1 (SSD) 465 GB

The virtual machine Oracle VirtualBox Manager framework is used to run the instant Contiki 2.7 operating system. Contiki contains the simulation environment and necessary tools to run the experiment.

4.2 Experiment Environment

Most researchers rely on simulation environments instead of real experiments in their research, and that is due to the availability and accessibility of the simulation software. 86% of the research relied on simulation software, while only 14% relied on real experiments, as shown in Figure 4.13. This significant difference in use is due to the lack of resources in the laboratories in addition to the high cost the researcher needs to pay. Simulation software offers unlimited resources in the experiment environment, which makes it easier to apply to any environment [38]. RPL is the optimal protocol for IPV6 applications, and evaluating RPL performance requires a large number of different IoT devices plus compatibility requirements and a large area laboratory with the ability to control and change the distance between IoT devices. Evaluating RPL performance needs a big network of IoT devices to allow a greater number of routes that IoT nodes can use to send data; H. Kim et al. The survey shows that most RPL objective functions evaluation research applied their experiment on IoT networks that consist of 50 IoT nodes or more. Figure 4.14 illustrates the number of nodes in the experiment according to the average number of research in the survey. Building IoT networks with a large number of nodes plus having a big area laboratory which can isolate the experiment from signal jamming and interference is hard to achieve. Due to the mentioned reasons and conditions, we decided to apply our experiment in a simulation software that can control and modify the experiment, such as network topology, number of IoT nodes, type of nodes, and ease of applying different tests [39].

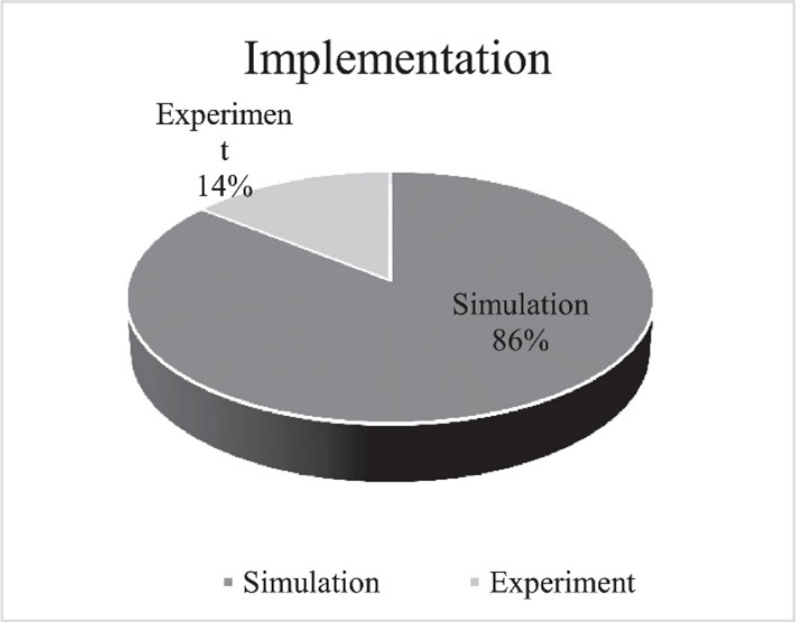


Figure 4.13: Distribution of simulated and real experiment in IoT research [39].

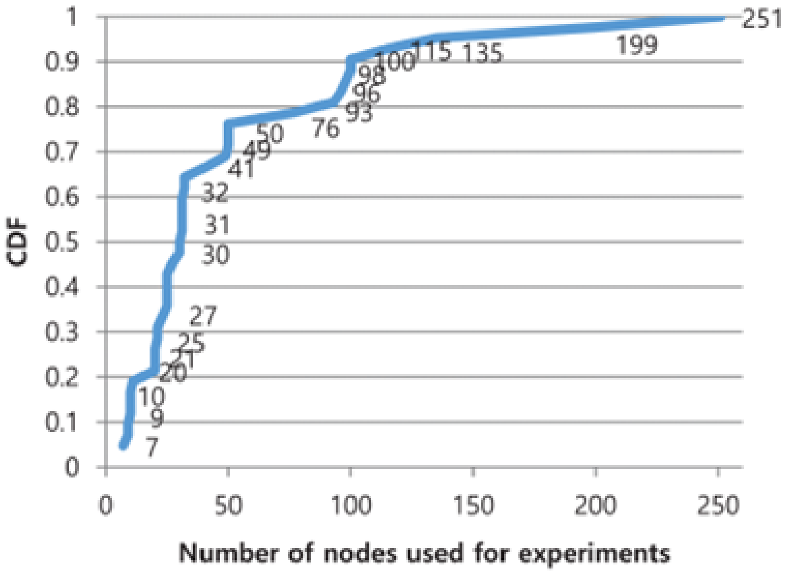


Figure 4.14: Number of IoT nodes used in research. CDF: Cumulative Distribution Function.

4.3 Contiki-OS

Contiki OS is a widely used IoT operating system in LLN networks since it is open-source software and supports most of the microprocessors and controllers which are considered the core of the IoT devices. It is a lightweight operating system that runs on a wide range of low-power and memory-constrained devices. Contiki OS provides a range of features that are specifically designed for IoT devices, including support for low-power wireless communication and routing protocols, such as RPL, 6LoWPAN, and CoAP.

One of the key features of Contiki OS is the support for network protocols, which allows IoT data packets to be forwarded to the command-and-control center or server via communication between the IoT devices. Contiki OS supports a range of network protocols, including IPv6, TCP, and UDP, as well as a range of application-layer protocols, such as HTTP [40].

Contiki OS also provides a range of tools and libraries that make developing applications for IoT devices easier. These include a range of sensor drivers, communication stacks, application-level libraries, and development tools, such as compilers, debuggers, and simulators. Contiki includes a range of memory management features that enable it to run on devices with limited memory. As a result, 81% of the research relied on Contiki OS as an operating system to evaluate and test RPL performance and properties, while 19% of the research relied on other IoT operating systems such as Tiny OS, Jsim, and Matlab as shown in Figure 4.15 [38].

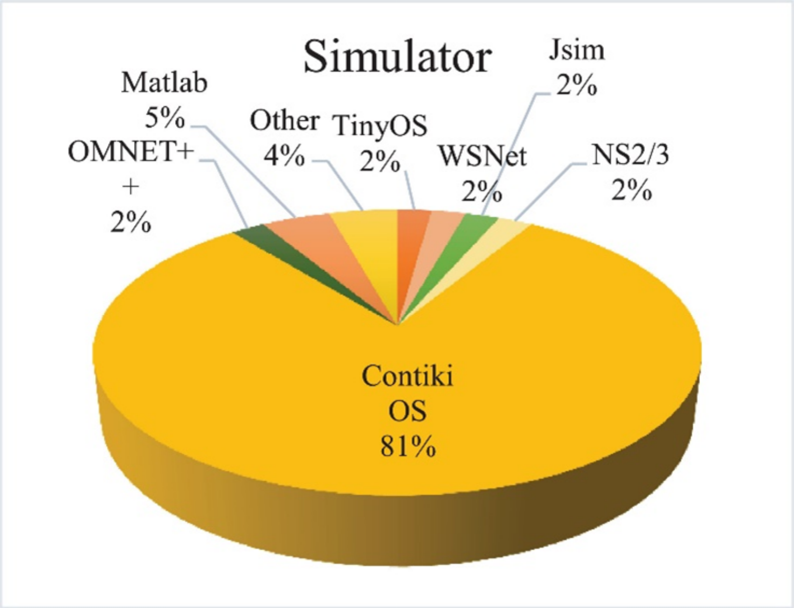


Figure 4.15: Operating system used in IoT research [38].

Overall Contiki OS provides simulation software or environment which is known as “Cooja”.

4.4 Cooja Simulator

Contiki OS provides The COOJA Simulator, which is an IoT network simulator designed for the Contiki OS. The main goal of the COOJA Simulator is to provide an extendable, usable, flexible, and scalable simulator that can be used during code development and testing phases, such as communication protocol performance testing. The simulator connects with Contiki, allowing simulated applications to run in a real Contiki system. By using this approach, any simulated application can then be run on a real IoT node unaltered. The simulator is designed to be used by both users and developers. It includes plugins and interfaces that allow interaction with the simulation environment, for example, control the simulation speed or watch all network traffic between the simulated nodes [41]. For all previous capability, flexibility, and features, Cooja was used by 63% of researchers to apply their research experiment, and the rest 37% used other simulation software such as NS-2/NS-3, WSNet, and OMNet as shown in the Figure below 4.16 [39].

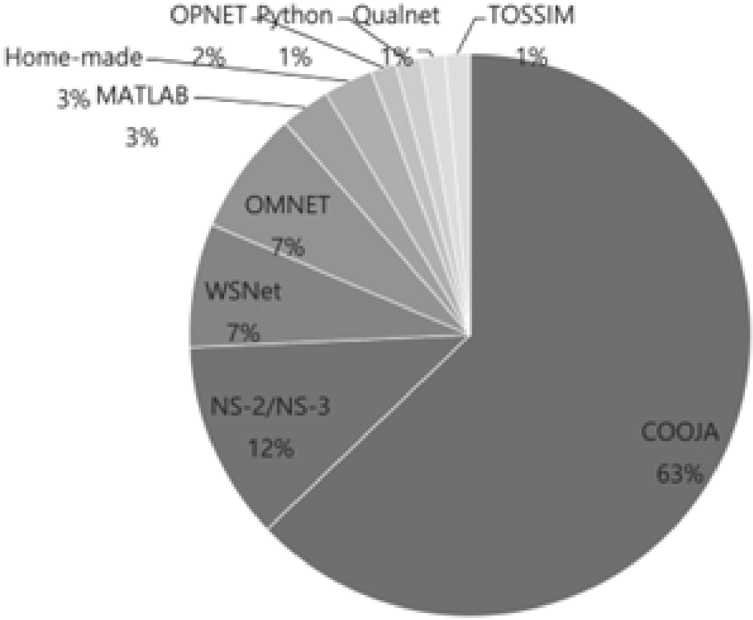


Figure 4.16: IoT network simulators used by researcher [39].

4.5 Building Network Topology

The network topology for this research is designed based on several factors. The network topology for this research is designed based on several factors. The first factor is IoT node density. Most research as mentioned build a network with a density of 50 nodes and more. The high density of nodes results in more paths and routes for IoT nodes to send data than low density, so a network with 100 client nodes is built to apply the RPL objective function performance experiment. The second factor is the node type as well as processing capability and memory size affect the node performance. The selected node type in this experiment is the skymote. The node distribution in the network is set as liner distribution, so most nodes are surrounded by eight neighbouring nodes in the signal coverage area, as shown in Figure 4.17. Another factor is node behaviour, so nodes should send data at different intervals. The 100 nodes are divided into four groups based on data sending time distributed randomly in the liner network topology. The sending time interval for node groups are 3, 5, 7, and 10 seconds. The different sending time intervals will resemble real-life IoT network data transmission. The network topology contains one root node (sink node) as type sky surrounded by six client nodes. The root node works to receive all data packets from client nodes in the topology network. The same network topology is used throughout the whole experiment to avoid effects caused by different network topologies.

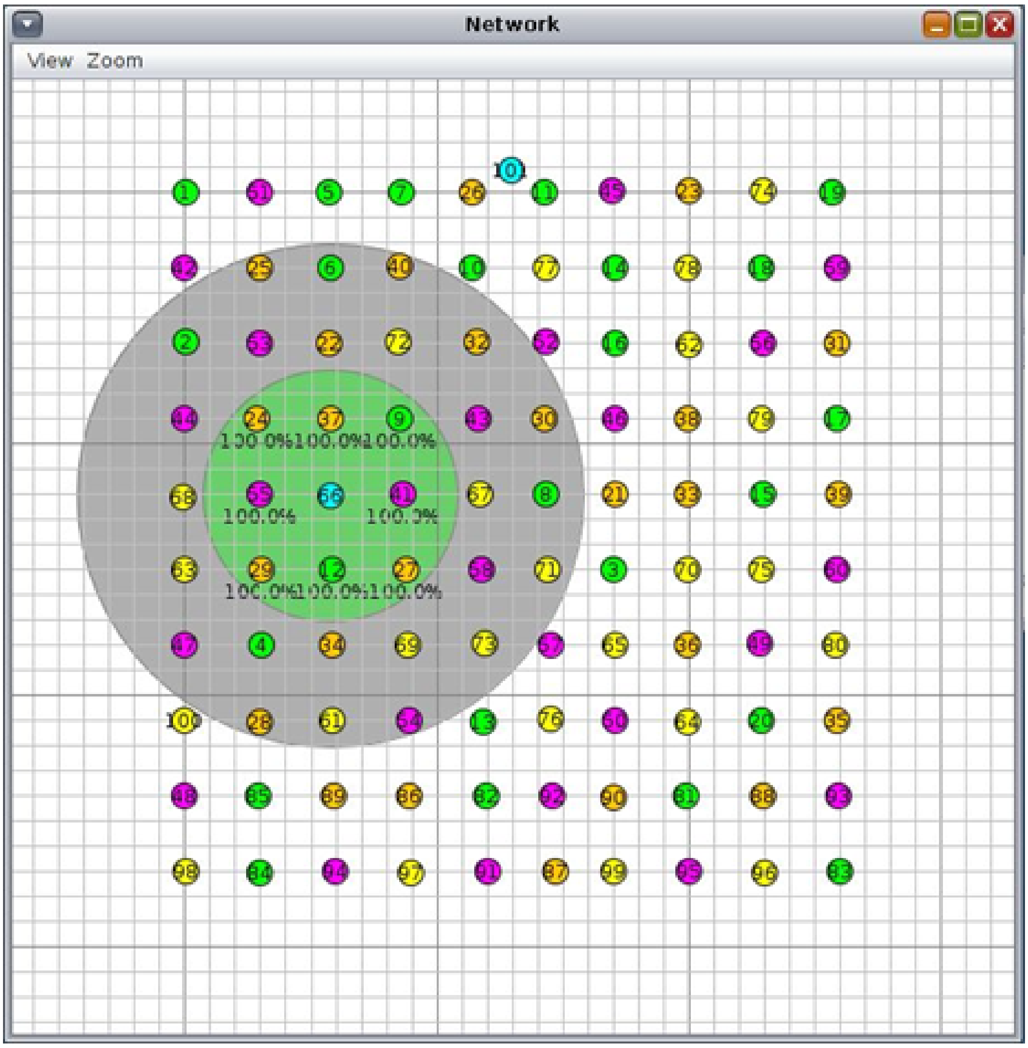
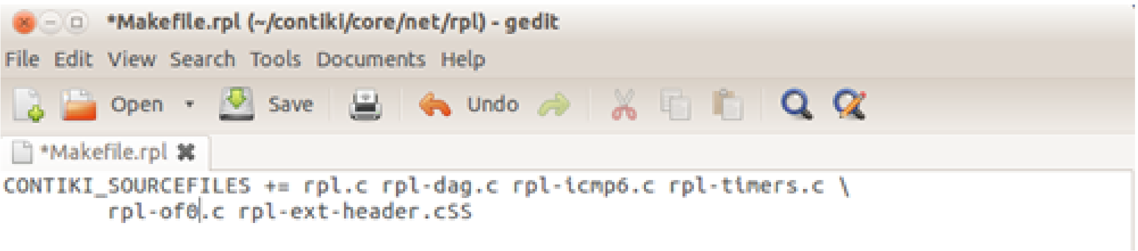


Figure 4.17: Network topology used in the experiment.

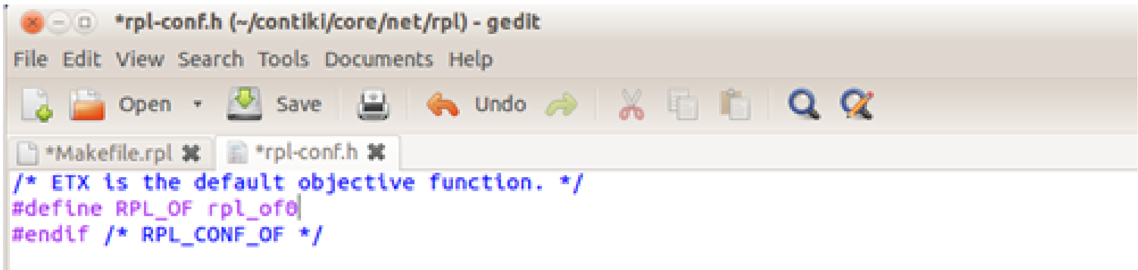
4.6 Objective function zero (OF0) implementation

The default objective function for RPL protocol is defined in RPL network configuration files which are rpl-conf.h and makefile.rpl as shown in Figure 4.18 and 4.19. The OF0 evaluation is based on network building time, total overhead, packet drop, total latency, and packet delivery ratio. The designated time to build the network is measured from the first DIO message sent by the root node until the last node joins the network. To generate the required logs to calculate network building time, the source code is modified to generate logs for each DIO sent and for each node that joined the network. The total overhead in the network or total sent control messages throughout the experiment is calculated by modifying the source code to generate a log for each control message DIO, DIS, and DAO send. The experiment relies on the application RPL-UDP which includes udp-server.c, which represents the root node application, and udp-client.c, which represents the client node application. For this experiment, the client node application is modified to let client nodes send data in different time intervals 3,5,7 and 10 seconds. In addition, to measure total power consumption and latency, the Powertrace application is added to the client node application. The experiment is performed on the mentioned network topology for a 30-minute time period, and all logs and network events are saved to the log file. Moreover, the experiment is replicated five times to guarantee the experiment results in accuracy and detect any failure.



```
CONTIKI_SOURCEFILES += rpl.c rpl-dag.c rpl-icmp6.c rpl-timers.c \
	rpl-of0.c rpl-ext-header.cSS
```

Figure 4.18: defining OF0 as default objective function for RPL in the Makefile.rpl.

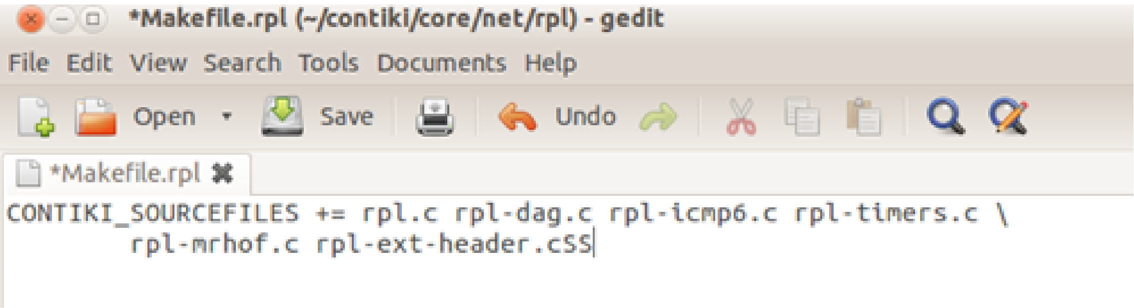


```
/* ETX is the default objective function. */
#define RPL_OF rpl_of0
#endif /* RPL_CONF_OF */
```

Figure 4.19: defining OF0 as default objective function for RPL in the rpl-conf.h.

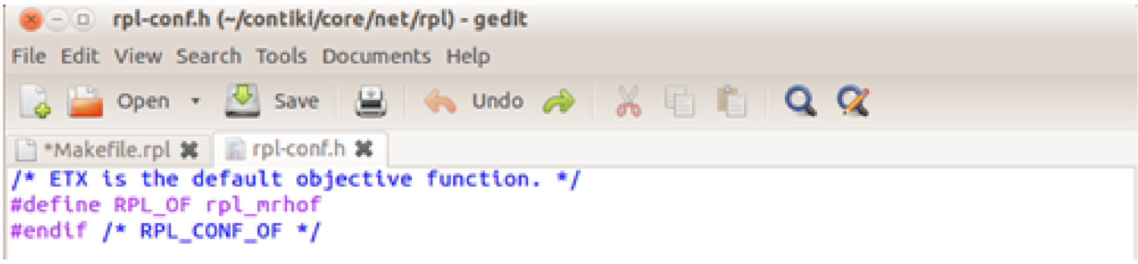
4.7 MRHOF implementation

To apply the MRHOF objective function, the same experiment is replicated using MRHOF as the default objective function by defining it in the RPL network configuration files rpl-conf.h and makefile.rpl as shown in Figure 4.20 and 4.21 After defining MRHOF as default objective function, root node and client node application files need to be recompiled to include the changes of the objective functions. The time required to build the IoT network is measured by using the same methods used for the OF0 implementation. The number of control messages exchanged within the network is calculated using the same method used for OF0 implementation. The experiment log file records all logs about packets sent through the network and control messages. The same root node and client node applications in OF0 implementation are employed in MRHOF implementation with the same sending time intervals and power measurement settings. The experiment is performed on the same network topology in OF0 implementation for the same time period, which is 30 minutes. Moreover, the experiment is replicated five times to guarantee the experiment results in accuracy and detect any failure, and a log file is saved for each replication.



```
CONTIKI_SOURCEFILES += rpl.c rpl-dag.c rpl-icmp6.c rpl-timers.c \
        rpl-mrhof.c rpl-ext-header.cSS
```

Figure 4.20: Defining MRHOF as the default objective function in the Makefile.rpl.



```
/* ETX is the default objective function. */
#define RPL_OF rpl_mrhof
#endif /* RPL_CONF_OF */
```

Figure 4.21: Defining MRHOF as the default objective function in the rpl-conf.h.

4.8 The new proposed EC-MRHOF objective function.

The IoT node in the network acts as a router and works to forward all received packets from other nodes, the node which is the preferred parent for other nodes, to the next node in the path to the network sink or root node. When a parent node receives or faces a high load traffic as the preferred parent for a large number of nodes due to the bad distribution of nodes in the network topology, this node uses its own power and system resources to manage the network traffic. As a result, node power and system resources will rapidly deplete, and this issue is known as the Hotspots. On the other hand, when a parent node receives huge traffic from other nodes that need more capability than the parent node, that will lead to a high packet drop rate. This issue is known as the Bottleneck problem and occurs in IoT nodes that are considered as the preferred parent for a large number of surrounding nodes, these nodes are busy all the time either by sending their own data or by forwarding data from other nodes in the network, and all data in the network passes through these nodes that lead to depletion of its resources faster than their capacity and disconnect from the network. Hotspot and Bottleneck problems show high depletion of resources, especially the power capacity. We assume that power consumption should play a primary role in path selection. On the other hand, this research experiment and other research show that MRHOF performs more efficiently than OF0; therefore, this thesis project will try to develop and improve MRHOF objective function by taking into account the total power consumption for the IoT node in node rank calculating function. The changes in the MRHOF metric is a percentage increment of the link cost according to the total power consumption of the node as shown in the following equation:

$$RankIncrease = ETX + (ETX + \frac{TCP}{BC})$$

$$Rank(N) = Rank(PN) + RankIncrease$$

- ETX: Expected Transmission Count (MRHOF ranking metric)
- TPC: Total Power Consumption
- BC: Battery capacity
- Rank(PN): Parent Node Rank
- Rank(N): Node Rank

The fraction $\frac{TCP}{BC}$ presents the percentage usage of the battery capacity and presents the increment in the rank in the improved MRHOF objective function.

4.8.1 EC-MRHOF Implementation

The new, improved EC-MRHOF objective function is defined as the default OF for the RPL protocol. The experiment in the new objective function is applied to the same network topology used in the OF0 evaluation experiment. The new OF is assessed based on the same factors used for other standardized OF: network building time, total overhead, packet drop, total latency, and packet delivery ratio.

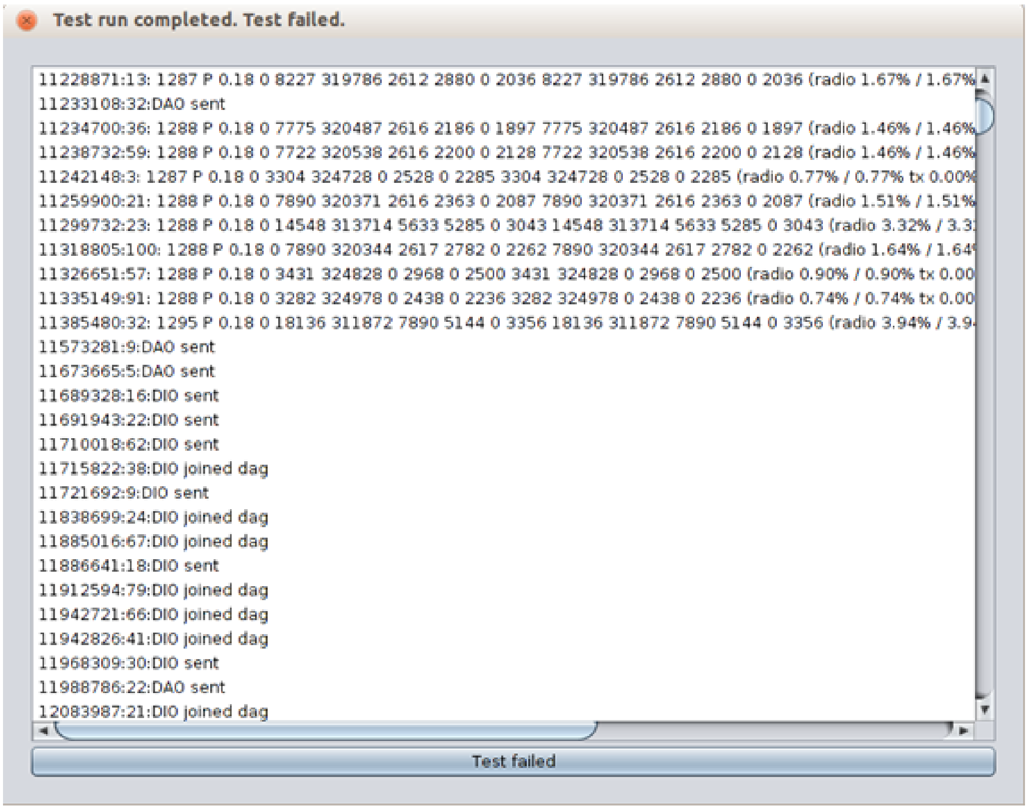
The time required to build the IoT network is measured by using the same methods used for the OF0 and MRHOF implementation. The number of control messages exchanged within the network is calculated using the same method used for OF0 and MRHOF implementation. The experiment log file records all logs about packets sent through the network and controls messages.

The same root node and client node applications in OF0 and MRHOF implementation are employed in the new objective function implementation with the same sending time intervals and power measurement settings. The experiment is performed for the same time of 30 minutes. Moreover, the experiment is replicated five times to guarantee the experiment results in accuracy and detect any failure, and log files are saved for each replication

4.8.2 Data Collection

Cooja simulator provides several plugins and tools that help to monitor the IoT network performance through the experiment duration, such as radio traffic, radio environment (UDGM), node output and graphical network performance statistics; Collect View, the graphical simulation operate for small network topology ranging from 5 to 10 nodes however for bigger network topology Cooja simulator runs slowly and lag after running the experiment for a short period.

Due to simulator limitations, the experiment is executed in a non-graphical user interface, command line, and script simulation. All experiment logs and events are recorded in log files and viewed at the end of the simulation as shown in Figure 4.22.

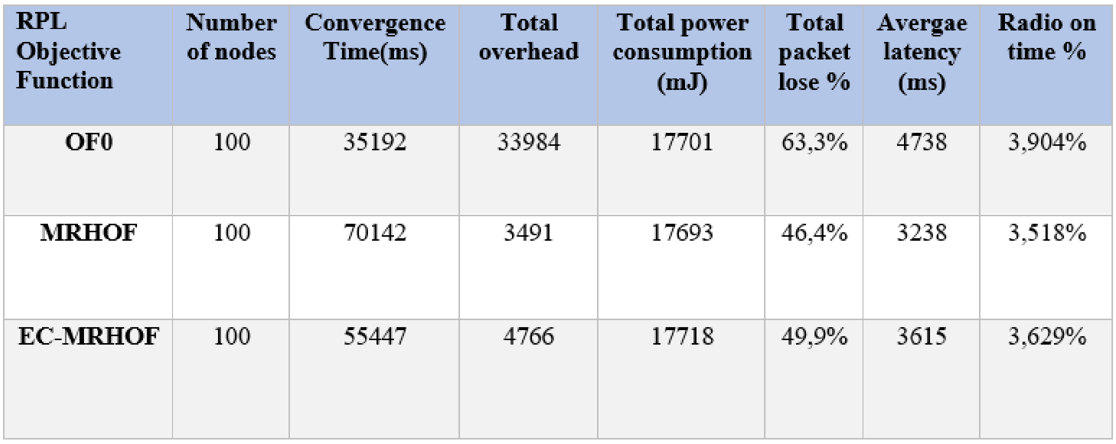


```
11228871:13: 1287 P 0.18 0 8227 319786 2612 2880 0 2036 8227 319786 2612 2880 0 2036 (radio 1.67% / 1.67%
11233108:32:DAO sent
11234700:36: 1288 P 0.18 0 7775 320487 2616 2186 0 1897 7775 320487 2616 2186 0 1897 (radio 1.46% / 1.46%
11238732:59: 1288 P 0.18 0 7722 320538 2616 2200 0 2128 7722 320538 2616 2200 0 2128 (radio 1.46% / 1.46%
11242148:3: 1287 P 0.18 0 3304 324728 0 2528 0 2285 3304 324728 0 2528 0 2285 (radio 0.77% / 0.77% tx 0.00%
11259900:21: 1288 P 0.18 0 7890 320371 2616 2363 0 2087 7890 320371 2616 2363 0 2087 (radio 1.51% / 1.51%
11299732:23: 1288 P 0.18 0 14548 313714 5633 5285 0 3043 14548 313714 5633 5285 0 3043 (radio 3.32% / 3.3
11318805:100: 1288 P 0.18 0 7890 320344 2617 2782 0 2262 7890 320344 2617 2782 0 2262 (radio 1.64% / 1.64
11326651:57: 1288 P 0.18 0 3431 324828 0 2968 0 2500 3431 324828 0 2968 0 2500 (radio 0.90% / 0.90% tx 0.00
11335149:91: 1288 P 0.18 0 3282 324978 0 2438 0 2236 3282 324978 0 2438 0 2236 (radio 0.74% / 0.74% tx 0.00
11385480:32: 1295 P 0.18 0 18136 311872 7890 5144 0 3356 18136 311872 7890 5144 0 3356 (radio 3.94% / 3.9
11573281:9:DAO sent
11673665:5:DAO sent
11689328:16:DIO sent
11691943:22:DIO sent
11710018:62:DIO sent
11715822:38:DIO joined dag
11721692:9:DIO sent
11838699:24:DIO joined dag
11885016:67:DIO joined dag
11886641:18:DIO sent
11912594:79:DIO joined dag
11942721:66:DIO joined dag
11942826:41:DIO joined dag
11968309:30:DIO sent
11988786:22:DAO sent
12083987:21:DIO joined dag
```

Figure 4.22: Non-GUI Simulation test script logs.

5 Results

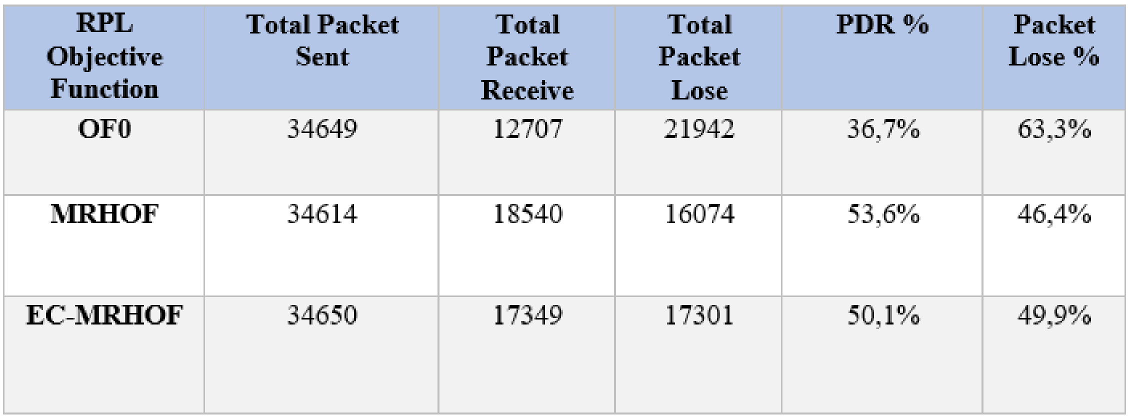
This chapter presents the results of conducting the three objective functions OF0, MRHOF and EC-MRHOF. The experiment results cover the output total time for building RPL network topology, total overhead, total power consumption, number of control messages, packet loss and total latency. Lastly, the results will be displayed in the form of a statistical graph. The presented results of the experiment are extracted from the log files using a Perl script.



RPL Objective Function	Number of nodes	Convergence Time(ms)	Total overhead	Total power consumption (mJ)	Total packet lose %	Avergae latency (ms)	Radio on time %
OF0	100	35192	33984	17701	63,3%	4738	3,904%
MRHOF	100	70142	3491	17693	46,4%	3238	3,518%
EC-MRHOF	100	55447	4766	17718	49,9%	3615	3,629%

Figure 5.23: RPL Objective Functions Evaluation Statistics.

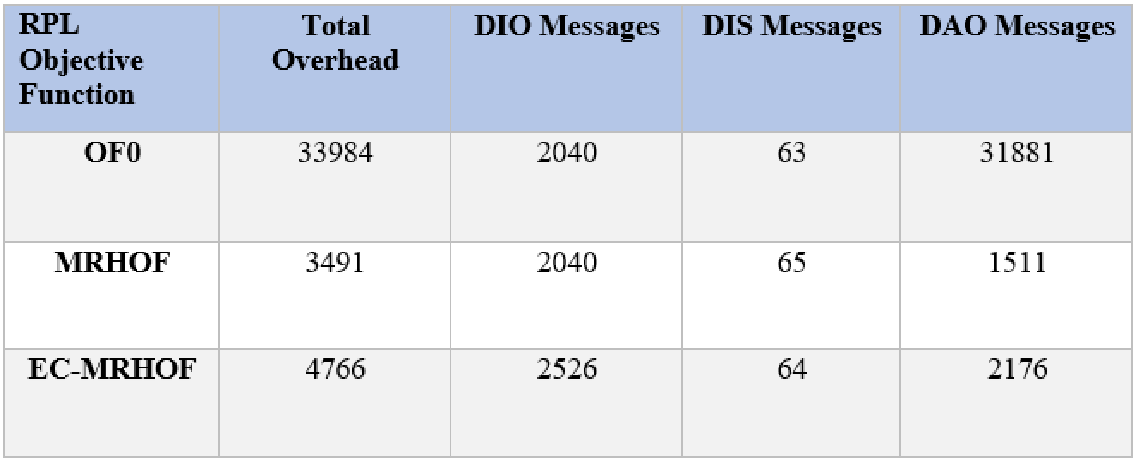
Figure 5.23 displays the results of executing the three RPL objective functions that includes network set up time, total overhead, total power consumption measured by millijoule(mJ) unit , percentage total packet lose, average latency measured by milliseconds (ms) and percentage of the radio on time.



RPL Objective Function	Total Packet Sent	Total Packet Receive	Total Packet Lose	PDR %	Packet Lose %
OF0	34649	12707	21942	36,7%	63,3%
MRHOF	34614	18540	16074	53,6%	46,4%
EC-MRHOF	34650	17349	17301	50,1%	49,9%

Figure 5.24: Packet Transmission Statistics.

Figure 5.24 shows statistics of packet transmission within the IoT network in different experiments. Which include the Total packet sent, total packet received, total packet lose, percentage of the packet Delivery Ratio (PDR) and percentage of packet lose.



RPL Objective Function	Total Overhead	DIO Messages	DIS Messages	DAO Messages
OF0	33984	2040	63	31881
MRHOF	3491	2040	65	1511
EC-MRHOF	4766	2526	64	2176

Figure 5.25: Network Overhead Statistics.

Figure 5.25 displays the measurement of total overhead in the network which presents the control messages sent by RPL protocol to build the network topology and define path or routes for the transmitted data.

6 Analysis

In this chapter, the results of testing the performance of three RPL objective functions will be analyzed and compared based on the performance criteria shown in Section 5. The final step of data analysis is to apply statistical testing on the result comparison to define performance differences between the three objective functions..

6.1 Convergence Time:

Based on the experiment´s results shown in Figure 5.23, the objective function zero take the shortest time to build the topology network which equal the half time MRHOF take and two-third EC-MRHOF take. OF0 relies on multi-hop count as a metric which does not require measurement of technical properties such as transmission cost to calculate the rank, so OF0 performs faster than other objective functions. Figure **??** shows the required time to build the network for the three objective functions measured in milliseconds (ms).

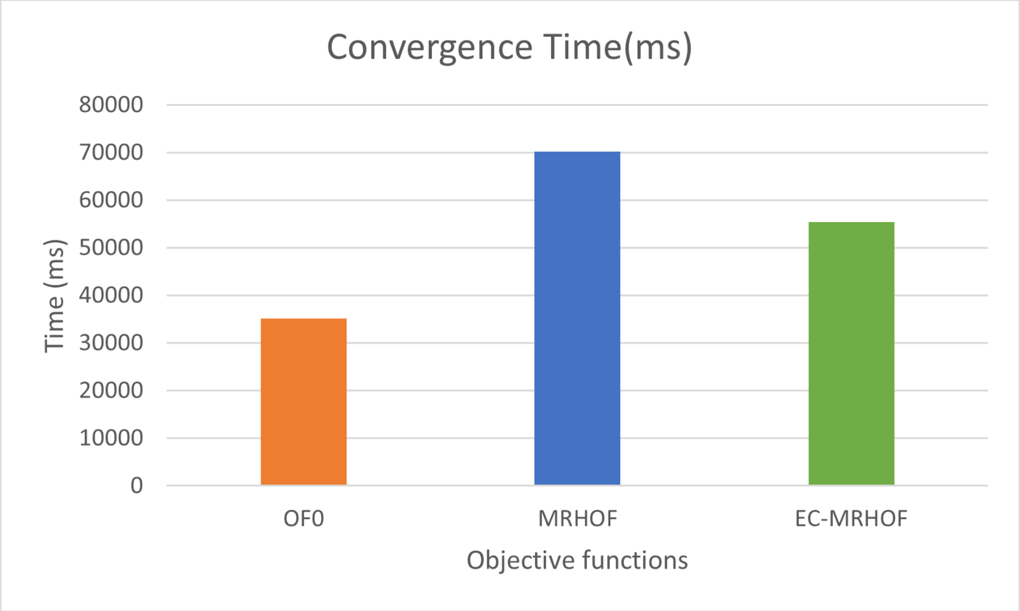


Figure 6.26: Network Setup Time for objective functions.

Figures 6.27 display the process of building the IoT network during the time for each Objective Function.

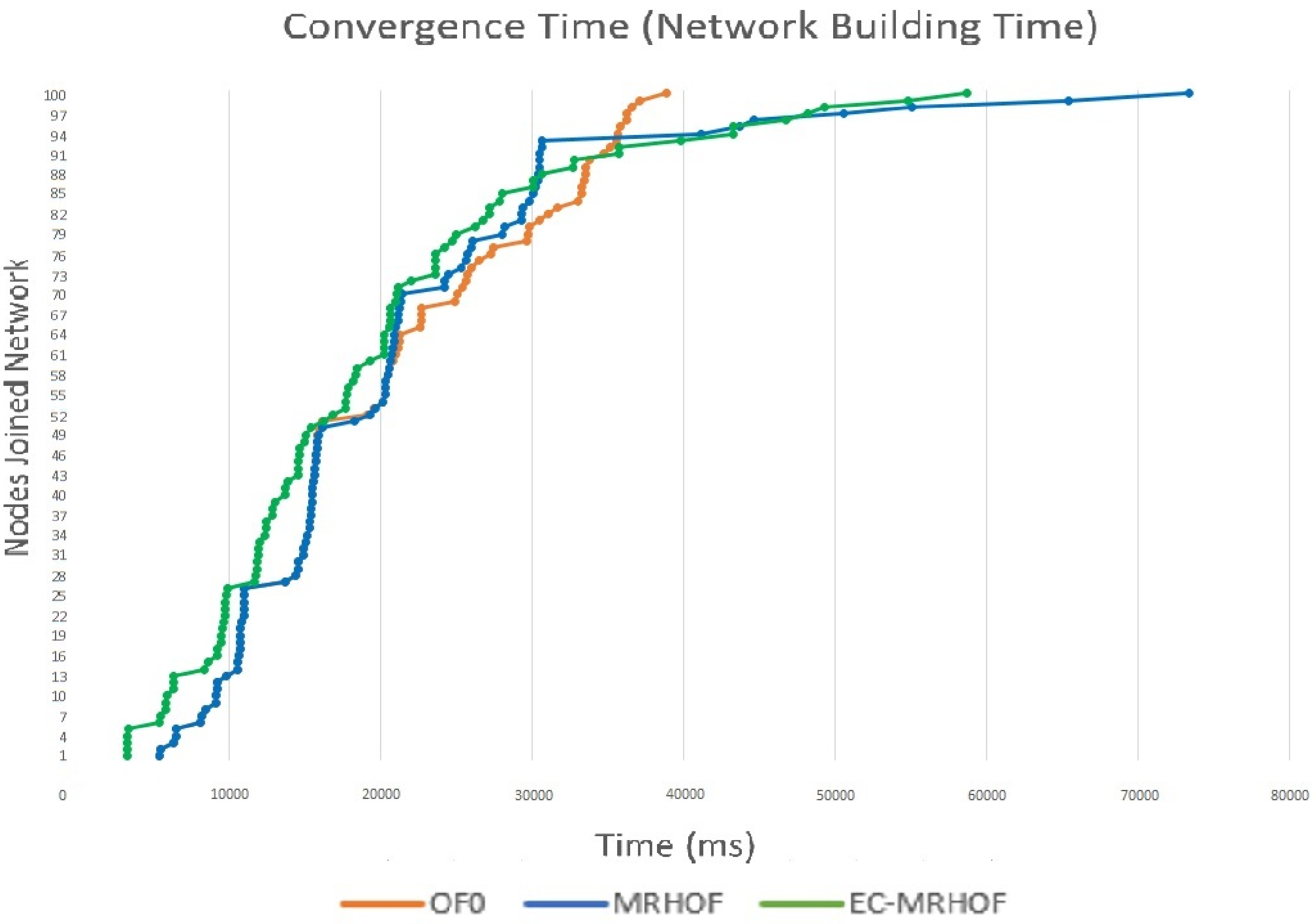


Figure 6.27: Network building time for the three Objective Functions.

6.2 Total Overhead

Based on the experiment´s results shown in Table 5.25, Objective Function Zero sends the highest number of control messages, which is approximately ten times as much as other objective functions. EC-MRHOF sent 36% more control messages than MRHOF. The vast amount of control messages sent by OF0 is caused mainly by DAO messages which are used to check the connectivity between nodes and to the sink nodes, which indicate instability in the link within the network. EC-MRHOF sends 32% more DIO messages than MRHOF. IoT nodes mainly use DIO messages to inform other nodes about them or any rank changes. This increase reflects the changes in ranking and selected paths, preferred parent, within the network.

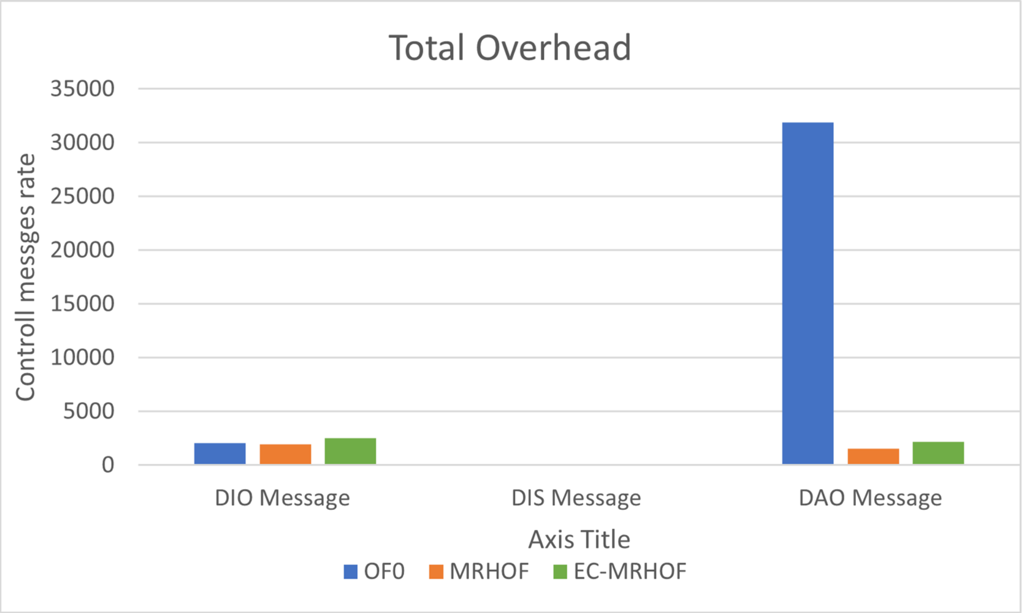


Figure 6.28: Network setup time for objective functions.

6.3 Packet Transmission

Based on the experiment´s results in Figure 5.24, Objective Function Zero shows the highest packet loss rate compared with the other objective functions due to the bad quality of selected paths based on the ranking metric. The difference between EC-MRHOF and MRHOF is slight, and MRHOF is appropriate in packet delivery. EC-MRHOF causes more packets to be lost due to selecting bad quality paths to balance power consumption between nodes.

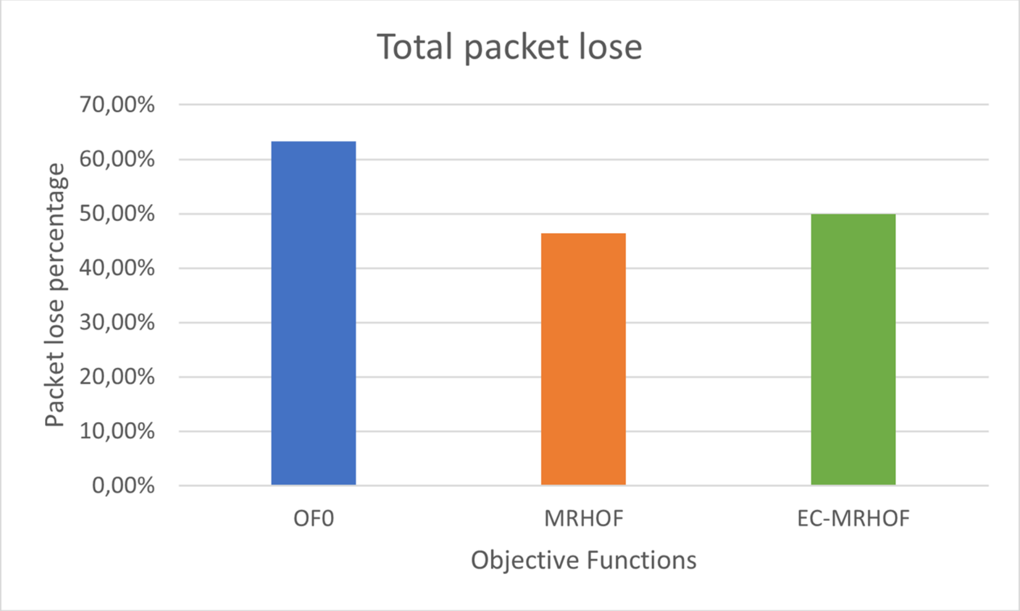


Figure 6.29: Total packet lose for objective functions.

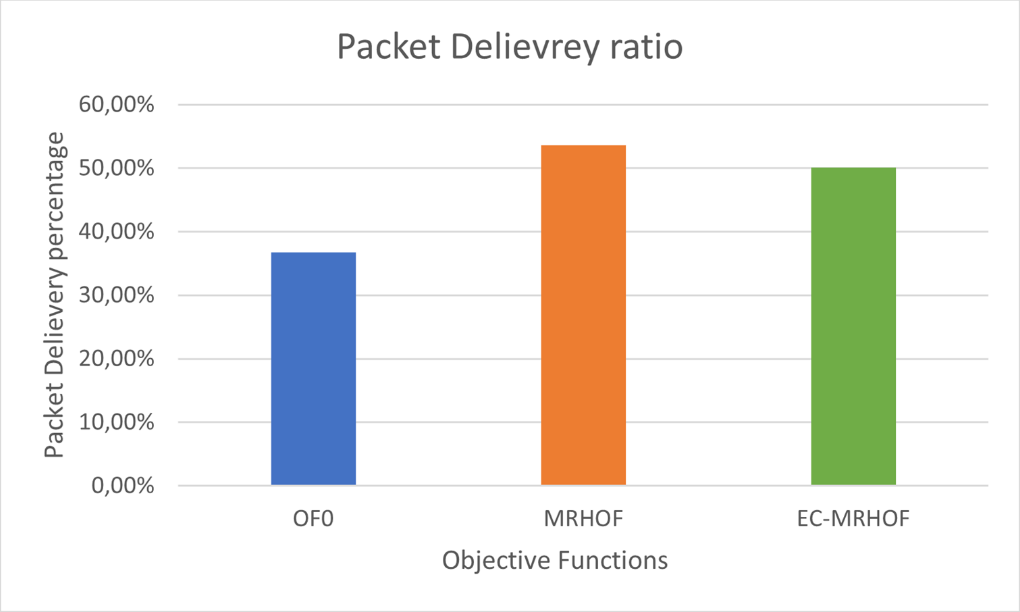


Figure 6.30: Total Packet Delivery Ratio for objective functions.

6.4 Latency

Based on the experiment´s results shown in Table 5.23, Objective Function Zero shows the highest latency average due to selecting the shortest routes based on hop count without considering the link cost. On the other hand, EC-MRHOF shows a slightly more latency rate than MRHOF. Due to this, EC-MRHOF selects high-cost links to balance power consumption between network nodes. MRHOF has the least latency rate among the other objective functions.

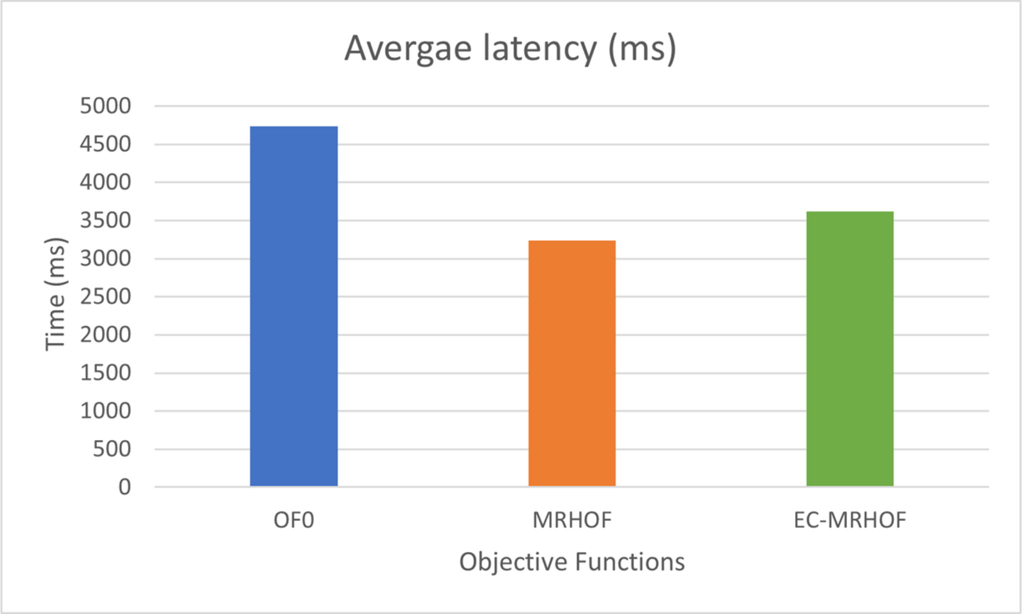


Figure 6.31: Average latency for the objective functions.

6.5 Radio on time

The node's radio on time represents the total time when the node sends or receives data or when the node is in an active mood. Based on the experiment's results in Table 5.23, the OF0 shows the highest percentage of radio on time. At the same time, EC-MRHOF and MRHOF show approximately the same percentage of radio on time.

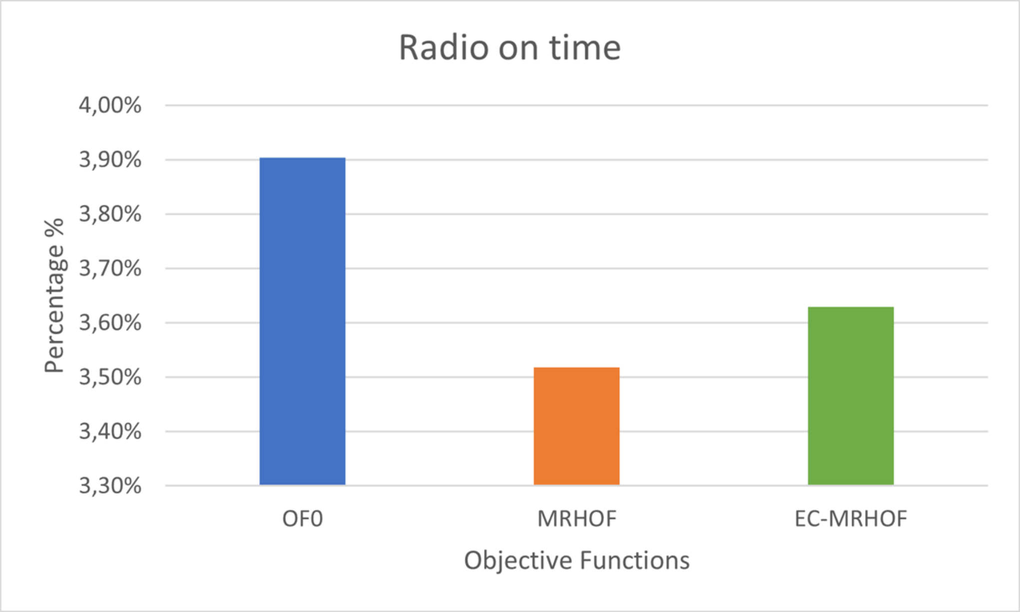


Figure 6.32: Radio on time for objective functions.

6.6 Power Consumption

Based on the experiment's results shown in Table 5.23, The total energy consumption for the network in different experiments shows similar energy consumption rates for the three objective functions. The RPL objective function differs in the consumption rate for different nodes in the network depending on the mechanism used for selecting routes in the network and the node's role in the data transmission process. The network topology is symmetrical with respect to the vertical axis passing through the root node, and every two opposite nodes from the two columns adjacent to the root node play the same role in the side of the network they belong to. Each node in these two columns does similar work and receives similar data traffic compared to the opposite node from the other column. The nodes in each pair of opposite nodes are highly selected as a preferred parents for the previous hop nodes in the path to the sink node. The node in each pair selected as a preferred parent will have a higher data traffic plus a higher power consumption. The new objective function EC-MRHOF should work to balance the power consumption for each node pair. Figure 6.35 shows the power consumption difference for each pair according to the three objective functions. OF0 show a slight difference for some pairs and a high difference for other compared with MRHOF and EC-MRHOF. However, OF0 uses different mechanisms and metrics to select the sink node path. So, the difference in power consumption will be random and not applicable to be compared to other objective functions. The EC-MRHOF objective function is an improvement of the MRHOF objective function. The EC-MRHOF shows less difference in power consumption for most pairs compared to the MRHOF, which means that EC-MRHOF works to balance power consumption between nodes, So EC-MRHOF is more efficient in managing power consumption in terms of selecting routes in the network than MRHOF.

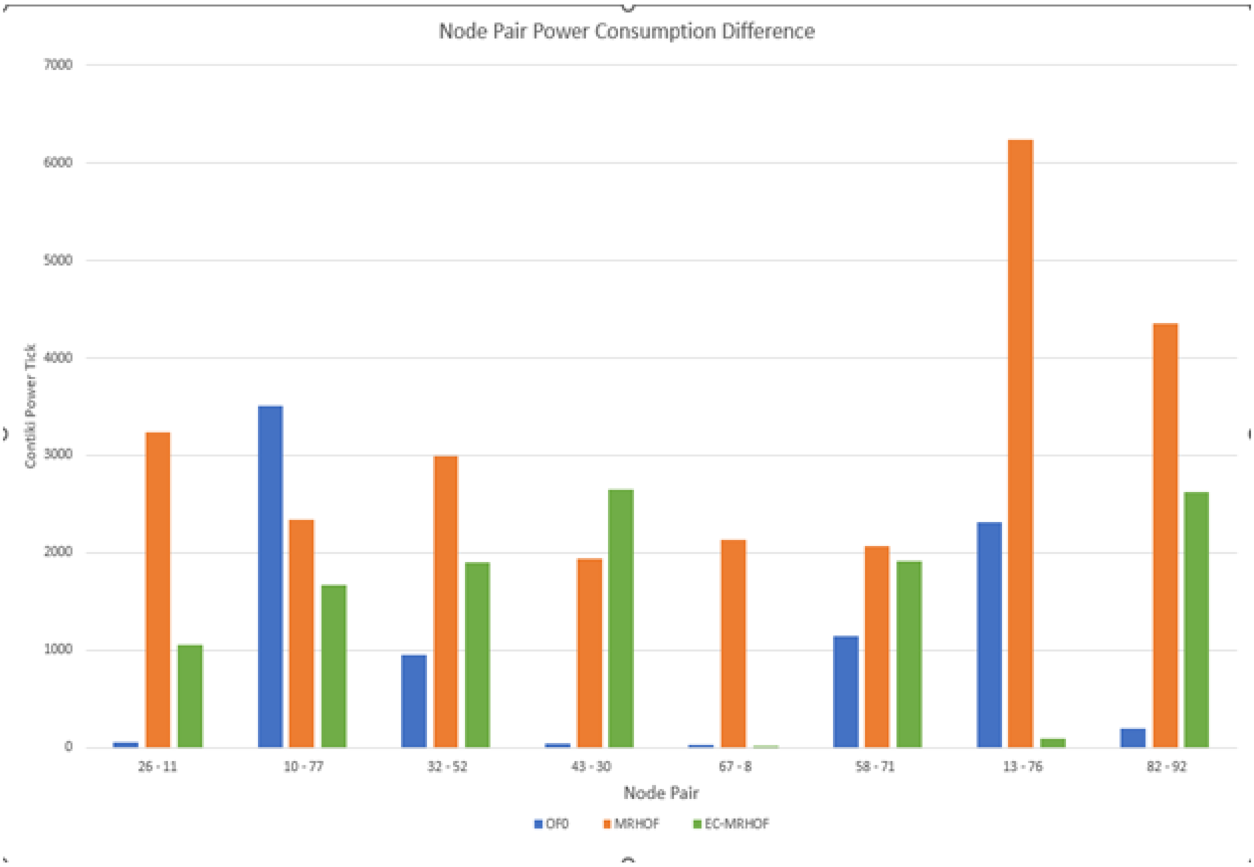


Figure 6.33: Graph illustrates the node pair power consumption difference based on the figure 6.34.

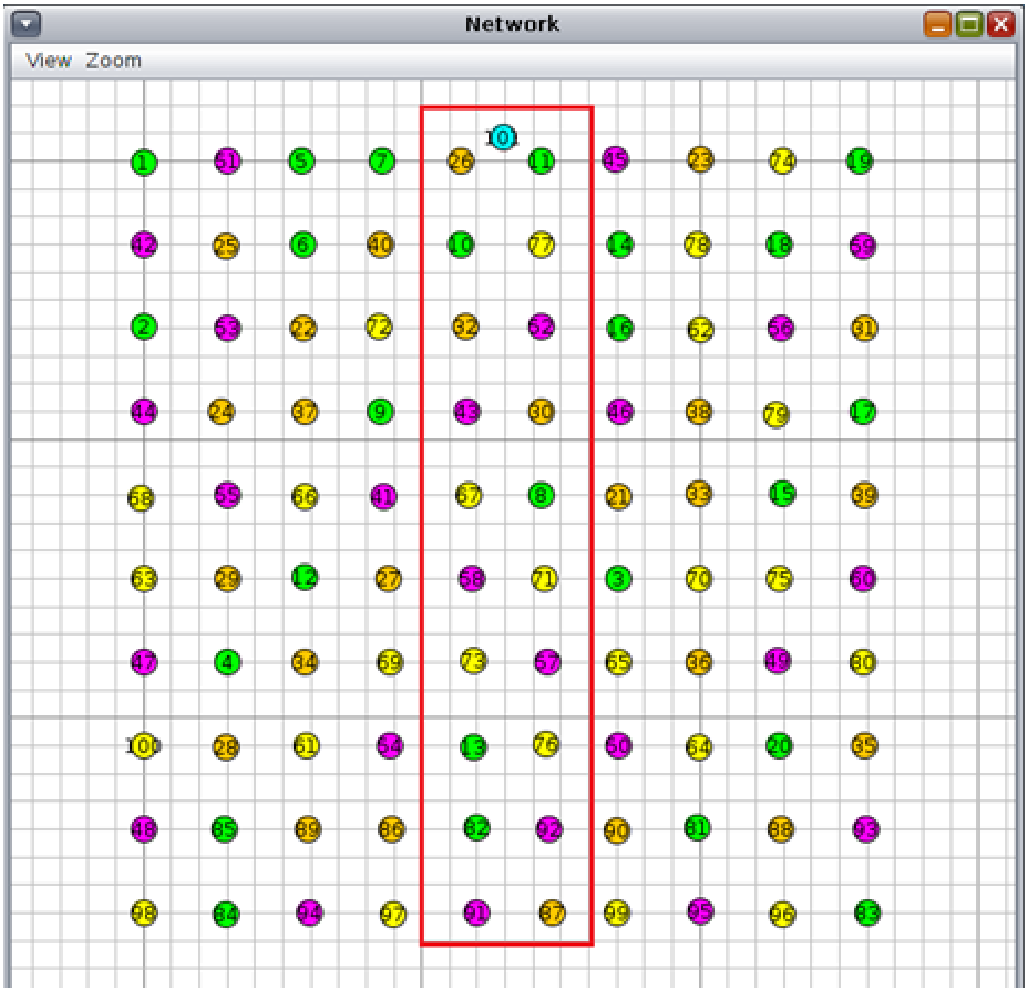


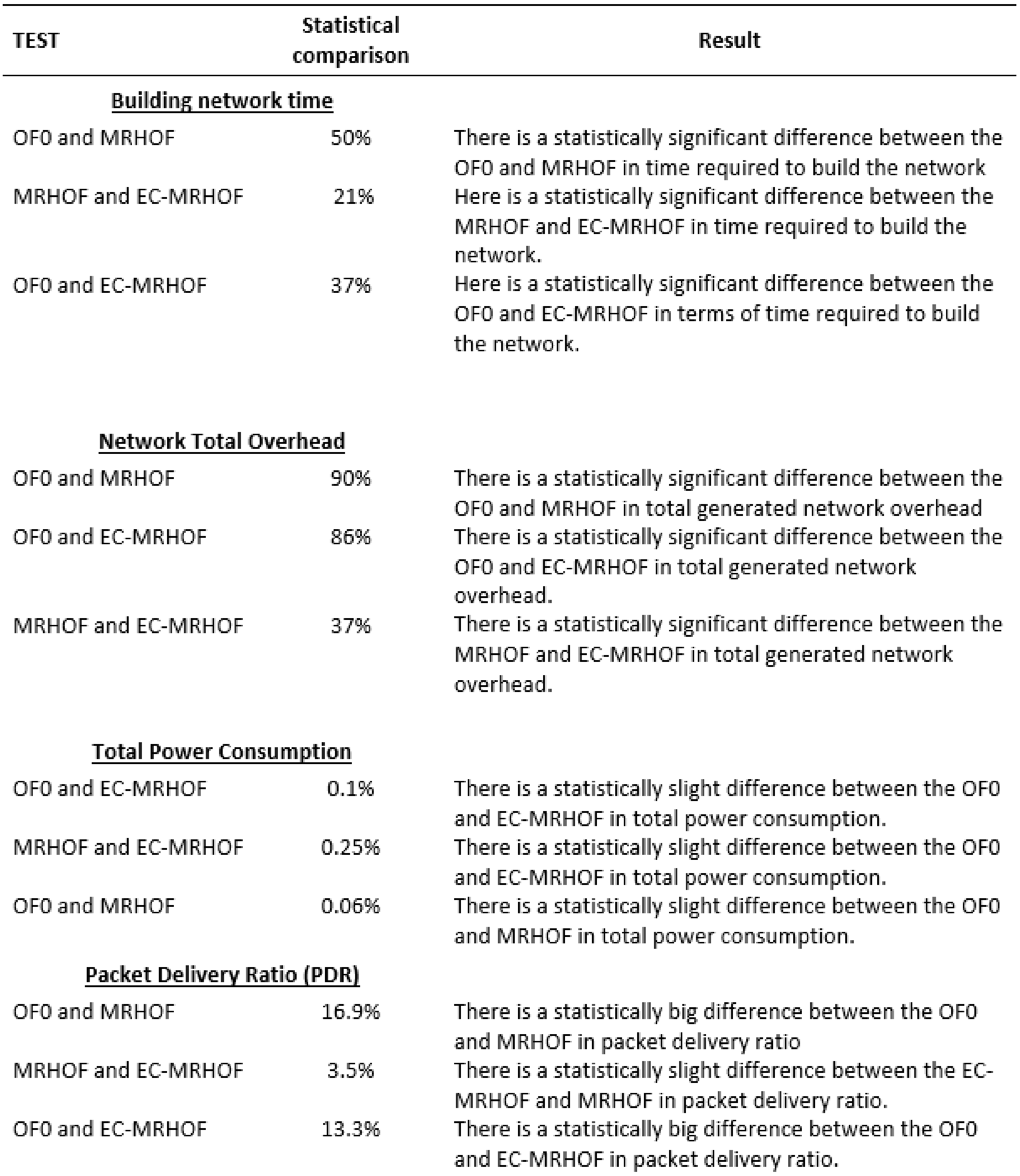
Figure 6.34: The chosen node pairs to be analyzed and compared in respect to power consumption.

Nodes	OF0	MRHOF	EC-MRHOF
26 - 11	56	3236	1057
10 - 77	3501	2338	1664
32 - 52	946	2998	1898
43 - 30	43	1945	2647
67 - 8	22	2131	12
58 - 71	1143	2072	1914
13 - 76	2312	6243	84
82 - 92	194	4358	2616

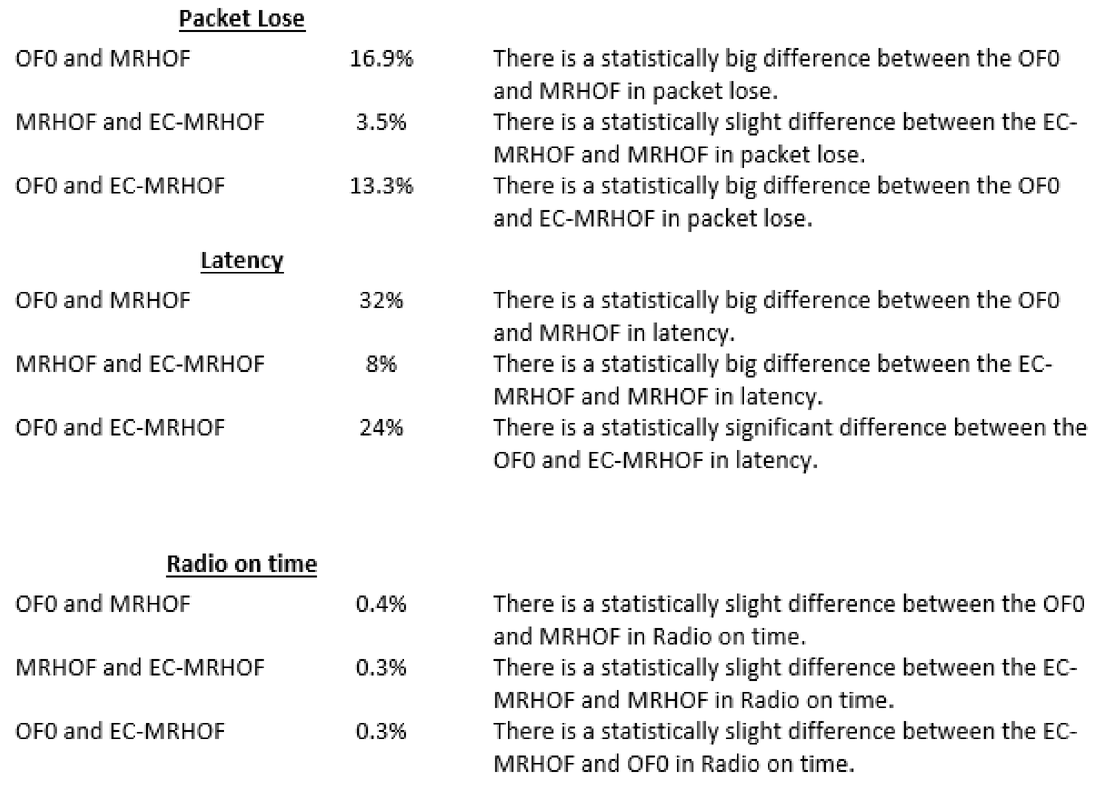
Figure 6.35: Total Power consumptions difference between the pair nodes for the three objective functions.

6.7 Statistical testing consideration

This research paper evaluates three different RPL objective functions OF0, MRHOF and EC-MRHOF. The experiment result comparison is between three values in terms of Convergence time, total power consumption, total overhead, packet delivery ratio, packet loss and radio on time. Regarding the number of compared values, the chosen statistical test approach is ANOVA.



TEST	Statistical comparison	Result
Building network time		
OF0 and MRHOF	50%	There is a statistically significant difference between the OF0 and MRHOF in time required to build the network
MRHOF and EC-MRHOF	21%	Here is a statistically significant difference between the MRHOF and EC-MRHOF in time required to build the network.
OF0 and EC-MRHOF	37%	Here is a statistically significant difference between the OF0 and EC-MRHOF in terms of time required to build the network.
Network Total Overhead		
OF0 and MRHOF	90%	There is a statistically significant difference between the OF0 and MRHOF in total generated network overhead
OF0 and EC-MRHOF	86%	There is a statistically significant difference between the OF0 and EC-MRHOF in total generated network overhead.
MRHOF and EC-MRHOF	37%	There is a statistically significant difference between the MRHOF and EC-MRHOF in total generated network overhead.
Total Power Consumption		
OF0 and EC-MRHOF	0.1%	There is a statistically slight difference between the OF0 and EC-MRHOF in total power consumption.
MRHOF and EC-MRHOF	0.25%	There is a statistically slight difference between the OF0 and EC-MRHOF in total power consumption.
OF0 and MRHOF	0.06%	There is a statistically slight difference between the OF0 and MRHOF in total power consumption.
Packet Delivery Ratio (PDR)		
OF0 and MRHOF	16.9%	There is a statistically big difference between the OF0 and MRHOF in packet delivery ratio
MRHOF and EC-MRHOF	3.5%	There is a statistically slight difference between the EC-MRHOF and MRHOF in packet delivery ratio.
OF0 and EC-MRHOF	13.3%	There is a statistically big difference between the OF0 and EC-MRHOF in packet delivery ratio.



Packet Lose

OF0 and MRHOF	16.9%	There is a statistically big difference between the OF0 and MRHOF in packet lose.
MRHOF and EC-MRHOF	3.5%	There is a statistically slight difference between the EC-MRHOF and MRHOF in packet lose.
OF0 and EC-MRHOF	13.3%	There is a statistically big difference between the OF0 and EC-MRHOF in packet lose.

Latency

OF0 and MRHOF	32%	There is a statistically big difference between the OF0 and MRHOF in latency.
MRHOF and EC-MRHOF	8%	There is a statistically big difference between the EC-MRHOF and MRHOF in latency.
OF0 and EC-MRHOF	24%	There is a statistically significant difference between the OF0 and EC-MRHOF in latency.

Radio on time

OF0 and MRHOF	0.4%	There is a statistically slight difference between the OF0 and MRHOF in Radio on time.
MRHOF and EC-MRHOF	0.3%	There is a statistically slight difference between the EC-MRHOF and MRHOF in Radio on time.
OF0 and EC-MRHOF	0.3%	There is a statistically slight difference between the EC-MRHOF and OF0 in Radio on time.

Figure 6.36: Statistical tesing comparision for the three main objective functions based on Anova approach.

7 Discussion

This research paper aimed to evaluate the performance of RPL standardized objective functions OF0 and MRHOF and to come up with a new objective function as a new metric. We tested the standardized objective functions in the IoT network with fixed experiment-dependent variables. Then created the new objective function EC-MRHOF as an improvement for MRHOF and tested the new objective function with the same IoT network topology, dependent variables, and experiment conditions. The evaluation of the three objective functions is compared in terms of different aspects, network setting time, total power consumption, packet overhead, Latency, PDR, and packet loss. The results´ comparison in Section 6 shows the performance difference between the three objective functions.

Concerning the results´ comparison, the MRHOF performed more efficiently than OF0; MRHOF has a higher packet delivery ratio than OF0 and a lower average of packet loss. MRHOF selects better routes in the network which improves the average network latency compared with OF0. Equally, MRHOF has less radio on time than OF0.MRHOF builds a more stable network topology than MRHOF and generates less network overhead. The average total power consumption was nearly equal for both objective functions. Therefore, MRHOF is more competent and stable in contrast with OF0.

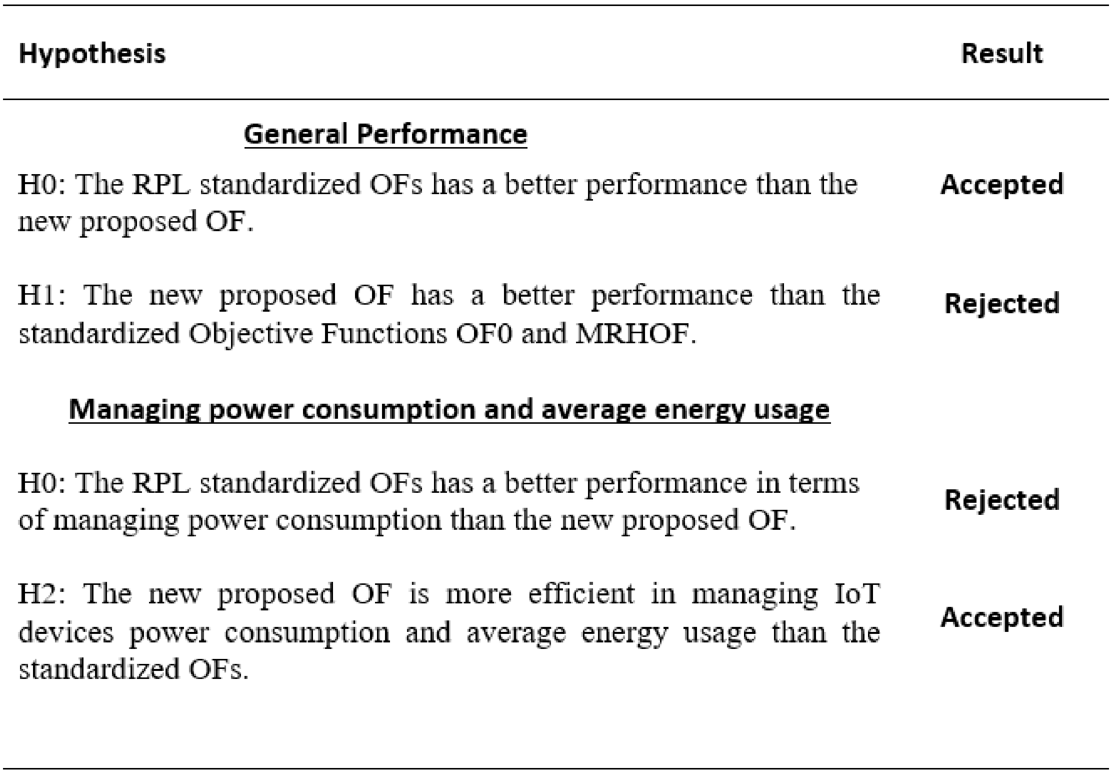
Jara et al[8].Conclude that MRHOF performs better than OF0 and is more competent and stable as RPL objective functions; our results correspond with Jara's related work research.
Qasem et al[7].Conclude that MRHOF and OF0 perform equally due to the low density of the experiment network topology, which will not show a significant performance difference; our results do not correspond with Qasim's related work research.

EC-MRHOF as an enhancement of the standardized MRHOF, performed better than OF0 in terms of most criteria, same as MRHOF. The average total power consumption was similar for both EC-MRHOF and OF0; thus, EC-MRHOF worked to balance power consumption between network nodes, while OF0 does not support any power consumption balancing. On the other hand, EC-MRHOF has a lower average of latency, packet-lose, network overhead and radio on time and a higher packet delivery ratio than OF0.It can be seen that EC-MRHOF is more efficient and competent than the standardized OF0.

EC-MRHOF is an improvement of MRHOF, and the result comparison shows the effects of this improvement. EC-MRHOF required 21% less time to build the IoT network; also, the average total power consumption was the same for both objective functions. EC-MRHOF generated more network overhead than MRHOF; this increase was in DIO and DAO messages with the same increase value. The rise of control messages is due to path changes by the objective function to balance the power consumption between the network nodes. Similarly, EC-MRHOF produced an 8% higher average of network latency than MRHOF due to selecting lower quality links by EC-MRHOF to achieve the power consumption balancing in the network. Based on statistical testing, EC-MRHOF had a slightly lower packet delivery ratio and marginally higher packet loss ratio than MRHOF, and this slight difference will not be considered, so we consider EC-MRHOF and MRHOF performed similarly with respect to packet delivery ratio and packet loss. IoT network nodes had a slight difference in radio on time, leading to consider both ob-

jective functions EC-MRHOF and MRHOF, which performed equally in regard to IoT node´s radio on time.
Based on the performance comparison results between RPL objective functions, the H0 hypothesis is accepted as it fulfilled the best performance terms and H1 hypothesis is rejected.

The total power consumption was the same for both objective functions, EC-MRHOF and MRHOF. The new objective function is designed to balance the power consumption between network nodes. According to Table 6.6, EC-MRHOF had lower power consumption differences for similar nodes compared with MRHOF due to the effect of the new objective function metric. Moreover, EC-MRHOF increases the node rank for high power consumption to adapt another node as the preferred parent and change the path for network packets. The EC-MRHOF had achieved power consumption balancing on the network nodes and contributed to the improvement of the IoT network lifetime.
Based on power consumptions comparison results between the RPL objective functions, the H2 hypothesis is accepted as the new proposed objective function (EC-MRHOF) is more efficient in managing power consumption and the H0 hypothesis is rejected in term of power consumption and average energy usage.



Hypothesis	**Result**
General Performance	
H0: The RPL standardized OFs has a better performance than the new proposed OF.	**Accepted**
H1: The new proposed OF has a better performance than the standardized Objective Functions OF0 and MRHOF.	**Rejected**
Managing power consumption and average energy usage	
H0: The RPL standardized OFs has a better performance in terms of managing power consumption than the new proposed OF.	**Rejected**
H2: The new proposed OF is more efficient in managing IoT devices power consumption and average energy usage than the standardized OFs.	**Accepted**

8 Conclusions and Future Work

8.1 Conclusion:

This thesis presented the evaluation of the standardized objective functions OF0 and MRHOF as well as the release of a new objective function EC-MRHOF. This research adopted the control experiment methodology to answer the following research objectives:

- Run RPL protocol and evaluate the performance of standardized OFs.
- Propose a new OF, evaluate the new OF, and compare the new OF's performance with standardized OFs.

Section 4.6 and 4.7 described the process of implementing and running the RPL protocol using the standardized objecvtive function. Section 6 and 7 showed the analyzing and comparison for the performance of standardized objective functions. Therefor the comparison clearly shows a significant performance difference between the standardized objective functions and proved that MRHOF is more efficient, capable, and reliable than OF0. So MRHOF should be nominated as the default objective function for the RPL protocol in different IoT sectors.

Section 4.8 illustrated the stages of building the new objective function EC-MRHOF. Moreover, Section 4.8.1 described the process of conducting and running the RPL protocol using EC-MRHOF objective function. Section 6 and 7 showed the analyzing and comparison of the new objective function EC-MRHOF with the standardized objective functions. EC-MRHOF showed a high ability in balancing the power consumption within the IoT network and improving the MRHOF in terms of power consumption. MRHOF and EC-MRHOF achieved better ranking and route selection in terms of different performance criterias.

As a result, MRHOF can be adopted as a default RPL objective function only for IoT networks and systems that include sustainable energy sources such as in industrial machines. EC-MRHOF proved to be better objective function for RPL protocol in IoT networks and systems that support mobility and work with limited power sources such as, medical system's equipment's and devices, agricultural monitoring systems and military IoT systems.

8.2 Future Work

The ranking mechanism uses one equation to merge two factors as an objective function's metric. Due to the time limitation, other merging mechanisms were not tested and evaluated, which could lead to more effective power consumption balancing.

- The fuzzy logic approach could be used to combine various factors to generate a new metric for the objective function.
- The idea of having two or more ranks based on various factors for the IoT nodes could be applicable to the new objective function instead of combining these factors in one node rank.
- EC-MRHOF could be evaluated for random network topology and networks with different densities.
- Test the new EC-MRHOF objective function improvement for the IoT network lifetime by estimating the network lifetime for each objective function.

References

[1] P. Gokhale, O. Bhat, and S. Bhat, “Introduction to iot,” *International Advanced Research Journal in Science, Engineering and Technology*, vol. 5, no. 1, pp. 41–44, 2018.

[2] H. Kharrufa, H. A. Al-Kashoash, and A. H. Kemp, “Rpl-based routing protocols in iot applications: A review,” *IEEE Sensors Journal*, vol. 19, no. 15, pp. 5952–5967, 2019.

[3] B. Stiller, E. Schiller, C. Schmitt, S. Ziegler, and M. James, “An overview of network communication technologies for iot,” *Handbook of Internet-of-Things*, vol. 12, 2020.

[4] M. Elappila, S. Chinara, and D. R. Parhi, “Survivable path routing in wsn for iot applications,” *Pervasive and Mobile Computing*, vol. 43, pp. 49–63, 2018.

[5] M. B. Yassien, O. Alzoubi, M. Shatnawi, and A. A. Rawashdeh, “Performance analysis of rpl objective functions,” in *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems*, 2019, pp. 1–6.

[6] P. Janani, V. Diniesh, and M. J. A. Jude, “Impact of path metrics on rpl’s performance in low power and lossy networks,” in *2018 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2018, pp. 0835–0839.

[7] M. Qasem, H. Altawssi, M. B. Yassien, and A. Al-Dubai, “Performance evaluation of rpl objective functions,” in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 2015, pp. 1606–1613.

[8] H. Lamaazi, N. Benamar, and A. J. Jara, “Study of the impact of designed objective function on the rpl-based routing protocol,” in *Advances in Ubiquitous Networking 2: Proceedings of the UNet’16 2*. Springer, 2017, pp. 67–80.

[9] L. Bassi, “Industry 4.0: Hope, hype or revolution?” in *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*. IEEE, 2017, pp. 1–6.

[10] J. Hagelbäck, “Degree projects in Computer Science” Method Overview,” https://coursepress.lnu.se/subject/thesis-projects/ieee-references/index.html, 2019.

[11] J. Lundberg, “2DV50E Controlled Experiment,” https://www.youtube.com/watch?v=2MqLFH7CalQ, April 2021.

[12] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[13] V. R. Basili, R. W. Selby, and D. H. Hutchens, “Experimentation in software engineering,” *IEEE Transactions on software engineering*, no. 7, pp. 733–743, 1986.

[14] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, and R. Alexander, “Rpl: Ipv6 routing protocol for low-power and lossy networks,” Tech. Rep., 2012.

[15] Y. B. Zikria, M. K. Afzal, F. Ishmanov, S. W. Kim, and H. Yu, “A survey on routing protocols supported by the contiki internet of things operating system,” *Future Generation Computer Systems*, vol. 82, pp. 200–219, 2018.

[16] Z. Shah, A. Levula, K. Khurshid, J. Ahmed, I. Ullah, and S. Singh, “Routing protocols for mobile internet of things (iot): A survey on challenges and solutions,” *Electronics*, vol. 10, no. 19, p. 2320, 2021.

[17] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, and R. Alexander, “Rpl: Ipv6 routing protocol for low-power and lossy networks,” Tech. Rep., 2012.

[18] H. Fotouhi, “Reliable mobility support in low-power wireless networks,” Ph.D. dissertation, Faculdade de Engenharia, Universidade do Porto, 2015.

[19] C. Gomez, J. Oller, and J. Paradells, “Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology,” *sensors*, vol. 12, no. 9, pp. 11 734–11 753, 2012.

[20] S. Safaric and K. Malaric, “Zigbee wireless standard,” in *Proceedings ELMAR 2006*. IEEE, 2006, pp. 259–262.

[21] U. Noreen, A. Bounceur, and L. Clavier, “A study of lora low power and wide area network technology,” in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE, 2017, pp. 1–6.

[22] A. Lavric and V. Popa, “Performance evaluation of lorawan communication scalability in large-scale wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[23] J. Ko, A. Terzis, S. Dawson-Haggerty, D. E. Culler, J. W. Hui, and P. Levis, “Connecting low-power and lossy networks to the internet,” *IEEE Communications Magazine*, vol. 49, no. 4, pp. 96–101, 2011.

[24] T. Salman and R. Jain, “Networking protocols and standards for internet of things,” *Internet of things and data analytics handbook*, pp. 215–238, 2017.

[25] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini, “Carp: A channel-aware routing protocol for underwater acoustic wireless networks,” *Ad Hoc Networks*, vol. 34, pp. 92–104, 2015.

[26] D. De Guglielmo, S. Brienza, and G. Anastasi, “Ieee 802.15. 4e: A survey,” *Computer Communications*, vol. 88, pp. 1–24, 2016.

[27] N. Sousa, J. V. Sobral, J. J. Rodrigues, R. A. Rabêlo, and P. Solic, “Eraof: A new rpl protocol objective function for internet of things applications,” in *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*. IEEE, 2017, pp. 1–5.

[28] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, and R. Alexander, “Rpl: Ipv6 routing protocol for low-power and lossy networks,” Tech. Rep., 2012.

[29] A. J. Witwit and A. K. Idrees, “A comprehensive review for rpl routing protocol in low power and lossy networks,” in *New Trends in Information and Communications Technology Applications: Third International Conference, NTICT 2018, Baghdad, Iraq, October 2–4, 2018, Proceedings 3*. Springer, 2018, pp. 50–66.

[30] I. S. Alsukayti and A. Singh, “A lightweight scheme for mitigating rpl version number attacks in iot networks,” *IEEE Access*, vol. 10, pp. 111 115–111 133, 2022.

[31] Y. Al Sawafi, A. Touzene, and R. Hedjam, “Hybrid deep learning-based intrusion detection system for rpl iot networks,” *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, p. 21, 2023.

[32] J. V. Sobral, J. J. Rodrigues, R. A. Rabêlo, J. Al-Muhtadi, and V. Korotaev, “Routing protocols for low power and lossy networks in internet of things applications,” *Sensors*, vol. 19, no. 9, p. 2144, 2019.

[33] O. Gaddour and A. Koubâa, “Rpl in a nutshell: A survey,” *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, 2012.

[34] E. Aljarrah, M. B. Yassein, and S. Aljawarneh, “Routing protocol of low-power and lossy network: Survey and open issues,” in *2016 International Conference on engineering & MIS (ICEMIS)*. IEEE, 2016, pp. 1–6.

[35] P. Thubert, “Objective function zero for the routing protocol for low-power and lossy networks (rpl),” Tech. Rep., 2012.

[36] O. Gnawali and P. Levis, “The minimum rank with hysteresis objective function,” Tech. Rep., 2012.

[37] N. Pradeska, W. Najib, S. S. Kusumawardani *et al.*, “Performance analysis of objective function mrhof and of0 in routing protocol rpl ipv6 over low power wireless personal area networks (6lowpan),” in *2016 8th international conference on information technology and electrical engineering (ICITEE)*. IEEE, 2016, pp. 1–6.

[38] H. Lamaazi and N. Benamar, “A comprehensive survey on enhancements and limitations of the rpl protocol: A focus on the objective function,” *Ad Hoc Networks*, vol. 96, p. 102001, 2020.

[39] H.-S. Kim, J. Ko, D. E. Culler, and J. Paek, “Challenging the ipv6 routing protocol for low-power and lossy networks (rpl): A survey,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2502–2525, 2017.

[40] F. Österlind, *A sensor network simulator for the Contiki OS*. Swedish Institute of Computer Science, 2006.

[41] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-level sensor network simulation with cooja,” in *Proceedings. 2006 31st IEEE conference on local computer networks*. IEEE, 2006, pp. 641–648.