

**Evaluating Routing Protocol for Low power and Lossy Networks (RPL)-Based Load  
Balancing Routing Algorithms in Internet of Things (IoT) Networks**

**By**  
**Zibuyisile Magubane**  
**(201123871)**

A dissertation submitted in fulfillment of the requirements for the degree of  
**Master of Science in Computer Science**

Faculty of Science, Agriculture and Engineering  
Department of Computer Science  
University of Zululand  
KwaDlangezwa  
RSA

Supervisor: Mr.Paul Tarwireyi

Co-supervisors: Prof Adnan Abu-Mahfouz

Prof Matthew Adigun

2022

# Abstract

The Internet of Things (IoT) is the network of different objects communicating different information in different scenarios. The networking objects are embedded with low-power devices that are responsible for collecting data in the physical environment and transmitting the data from one point to another. Routing is a mandatory factor in low-power devices to improve data transmission due to the volume of data transmitted in an IoT network. The Internet engineering task force designed an IPv6 routing protocol for low-power and lossy network (RPL) to govern data transmission in low-power devices. However, RPL fails to transmit data effectively in large IoT networks because it does not balance the load distribution during data transmission. Load balancing is the factor that enables the data to be distributed effectively among the IoT devices until they reach the destination. Several authors proposed load-balancing routing algorithms for RPL which were evaluated in different network scenarios. The proposed RPL-based load balancing routing algorithms offers partial load balancing and were evaluated in non-standard network areas and network size. Thus, it is challenging to identify an effective RPL-based load-balancing routing algorithm. To find the effective RPL-Based load balancing routing algorithms for IoT networks, we proposed three RPL-based load-balancing routing algorithms for the IoT network namely: Enhanced Context aware and load balancing routing algorithm for RPL (ENCLRPL), Buffer occupancy load balancing for RPL (BLRPL) and Enhanced ETXPC-RPL(EN-ETXPC-RPL). The design science research method (DSRM) was adopted to conduct this study. The algorithms were developed in the Contiki operating system and demonstrated in a simulation environment to find their effectiveness in an IoT network. The performance of the RPL-based load-balancing routing algorithms was evaluated based on reliability and stability metrics in different network sizes. The results obtained indicate that BLRPL is the most effective routing algorithm for IoT networks with the maximum of 96% of packet delivery ratio, 0.16 ms of network delay and 1.0 mW power consumption.

# Acknowledgements

Firstly, I want to thank God for being with me throughout this journey. I would also like to express my utmost gratitude to my supervisors Mr. P. Tarwireyi, Prof A. Abu-Mahfouz and Prof M.O Adigun for their support. Special thanks to Prof. P Mudali and the wireless mesh network cluster for continuous support and motivation. I am thankful for the CSIR funding on this project. I thank my Mother for her support and taking care of my son whilst I worked on my research, may God bless her.

# Contents

Abstract.....	i
Acknowledgements.....	ii
List of Tables .....	v
List of figures.....	vi
List of acronyms .....	vii
List of Publications .....	viii
Chapter 1.....	1
Introduction .....	1
1.1 Overview .....	1
1.2 Problem Statement.....	3
1.3 Research Questions .....	4
1.4 The Research Aim .....	4
1.5 The rationale of the Study .....	4
1.6 The contribution of the study .....	5
1.7 Organization of the dissertation .....	7
Chapter 2.....	9
Background of the study.....	9
2.1 IoT .....	9
2.2 Protocol stack.....	13
2.3 IoT Routing protocols.....	15
2.3.1 MQTT.....	15
2.3.2 CoAP.....	16
2.3.3 CARP .....	17
2.3.4 LOADng .....	17
2.3.5 RPL Overview .....	18
2.4 Load balancing .....	25
2.5 Summary .....	26
Chapter 3.....	28
A Review of RPL-Based Load Balancing approaches.....	28
3.1 Overview .....	28
3.2 Single Path RPL-based load balancing Routing Algorithms.....	29
3.3 Multipath RPL-based Load balancing routing algorithms.....	49

3.4	Summary .....	58
Chapter 4.....		60
Research Methodology .....		60
4.1	Problem identification and Motivation.....	62
4.2	The objective of the solution .....	63
4.3	Design and development .....	63
4.3.1	Contiki Operating System .....	63
4.4	Demonstration .....	74
4.5	Evaluation .....	80
4.6	Communication.....	82
4.7	Limitations of the research methods.....	83
4.8	Summary .....	83
Chapter 5.....		86
Results and Discussion .....		86
5.1	Results.....	86
5.2	Discussion.....	95
5.3	Recommendations .....	106
Chapter 6.....		109
Conclusion and Future Work .....		109
6.1	The research summary.....	109
6.2	Future Work.....	111
Bibliography .....		113

# List of Tables

Table 3.2 1: Single path RPL-Based Load balancing routing algorithms .....	42
Table 3.3 1: The summary of multiple path load balancing algorithms for RPL.....	55
Table 4.1: Design Science Research Methodology (Peffer et al 2007).....	61
Table 4.3. 1: RPL files in the Contiki operating system.....	66
Table 4.3. 2: RPL APIs used to create the new objective function CAO.....	67
Table 4.4.1: Parameters for Simulation.....	80

# List of figures

Figure 2. 1: IoT applications.....	10
Figure 2.2. 1: Contiki protocol stack (Seliem 2011).....	13
Figure 2.3 1: MQTT operational Diagram.....	16
Figure 2.3.4.1:LOADng protocol operation (Sobral et al 2019).....	18
Figure 2.5.5.1:Basic RPL operation with single DODAG in a single instance.....	20
Figure 2.5.3.1: Trickle operation.....	24
Figure 3.1: Different categories of RPL-based load balancing approaches.....	29
Figure 4.1: DSRM process Model which is an objectives-centered approach. ....	62
Figure 4.3.1.1Communication components in Contiki Operating System (Sebastian and Sivagurunathan 2018).....	65
Figure 4.3.1.2: RPL files in Contiki 2.7 with the rpl-caof objective function.....	67
Figure 4.3.1.3: BLRPL flow diagram.....	71
Figure 4.3.1.4: RPL files in Contiki 2.7.....	73
Figure 4.4.1: Cooja network simulation of 10 nodes randomly distributed in the network area..	77
Figure 4.4.2: Collect view plugin. ....	78
Figure 4.4.3: Wireshark packet analyzer.....	78
Figure 5.1.1: PDR of RPL-based load balancing routing algorithms in IoT network.....	87
Figure 5.1.2: Power consumed by the controller in the network.....	88
Figure 5.1.3:Power consumed in a sleep mode.....	89
Figure 5.1.4: Power consumed to listen on the transmission medium in the network. ....	90
Figure 5.1.5:Transmission power of Enhanced load balancing based on RPL. ....	91
Figure 5.1.6: Power Total power Consumed during data exchange.....	92
Figure.5.1.7: Delay in the network. ....	93
Figure 5.1.8: Total routes used for communication.....	94
Figure 5.1.9: Control overhead of RPL-load balancing algorithms.....	95
Figure 5.2.1: Packets successfully delivered to the destination in the IoT network.....	98
Figure 5.2.2: CPU Power in IoT network.....	99
Figure 5.2.3: Power consumed while devices are in sleep mode.....	100
Figure 5.2.4: Power consumed to listen the incoming packets.....	101
Figure 5.2.5: Power consumed to transmit data.....	102
Figure 5.2.6: Total power consumed in IoT network. ....	103
Figure 5.2.7: Network delay in different network sizes.....	104
Figure 5.2.8: Routes utilized during data transmission in different network sizes.....	105
Figure 5.2.9: Control Messages generated in the network.....	106

# List of acronyms

IoT	Internet of Things
6LoWPAN	IPv6 over Low power Wireless Personal Area Network
WSN	Wireless Sensor Networks
LLNs	Low power and Lossy Networks
M2M	Machine to Machine
IETF	Internet Engineering Task Force
IPv6	Internet Protocol version six
RPL	Routing Protocol for Low power and lossy networks
MRHOF	Minimum Rank with Hysteresis Objective Function
OF0	Objective Function Zero
ICMP	Internet Control Messages Protocol
DODAG	Destination-Oriented Directed Acyclic Graph
DAG	Directed Acyclic Graph
DIO	DODAG Information Object
DIS	DODAG Information Solicitation
MQTT	Message Queuing Telemetry Transport
HTTP	Hypertext Transfer Protocol
COAP	Constrained Application Protocol
CARP	Channel Aware Routing Protocol
LOADng	Lightweight On-demand Ad-hock Distance vector protocol-Next generation
UDGM	Unit Disk Graph Medium

# List of Publications

- Z. Magubane, P. Tarwireyi, A. M. Abu-Mahfouz and M. O. Adigun, "RPL-Based on Load Balancing Routing Objective Functions for IoTs in Distributed Networks," *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2019
- Z. Magubane, P. Tarwireyi, A. M. Abu-Mahfouz and M. O. Adigun, "Performance Evaluation of ETXPC-RPL Routing Algorithm in IoT Network," *2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2020.
- Z. Magubane, P. Tarwireyi, A. Abu-Mahfouz and M. Adigun, "Extended Context-Aware and Load Balancing Routing Protocol for Low power and Lossy Networks in IoT networks (ECLRPL)," *2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2021.
- Z. Magubane, P. Tarwireyi, A. M. Abu-Mahfouz and M. O. Adigun," Buffer Occupancy Load Balancing routing algorithm for RPL in IoT network (BLRPL)" **Accepted for publication** in the *Journal Of theoretical and Applied Information Technology (E-ISSN 1817-3195 / ISSN 1992-8645)*, 2023.

# Chapter 1

## Introduction

### 1.1 Overview

The Internet of Things (IoT) changes the way we live by allowing the connection and communication of devices without human intervention. IoT is adopted in almost all industries including engineering, health, agriculture, and so on. As industries adopt IoT, the number of IoT devices connected to the internet increases exponentially. Nearly 80 billion gadgets will be online in 2025, predicts Vernon Turner of IDC (Kanellos, 2016). These gadgets produce a flood of data and disseminate crucial environmental information. The data produced lead to high traffic in the network and create difficulties in data transmission. Routing data between IoT devices helps increase data transfer in the IoT network by providing a dependable path between devices. Several routing protocols were proposed by the Internet Engineering Task Force (IETF) for data transport in low-power devices including the Cluster Optimized Routing Protocol (CORP) (Jothikumar et al 2019), the Channel Aware Routing Protocol (CARP) (Basagni *et al.* 2015), and the Routing Protocol for Low power and lossy Networks(RPL) (Gaddour and Koubâa 2012). The protocols were assessed as follows:

- (1) CORP routing protocol is highly suitable for routing in low network sizes. In a large network, it results in routing loops and high node unreachability when the cluster head is out of power.
- (2) CARP protocol is routing data packets focusing on the channel's bandwidth and ignoring the delay of the packets thus making CARP unfit for IoT applications.
- (3) The RPL protocol performs routing based on a set of objective functions and routing metrics to conserve the power of the nodes and transmit packets at minimum latency. In low network traffic RPL transmit the data effectively. Therefore, RPL was standardized for Low power and Lossy Networks and adopted for IoT later. In IoT networks, RPL faces many challenges such as severe packet loss due to data heterogeneity, congestion due to poor data distribution, and short network lifetime. RPL has all these problems because there is a lack of load balancing mechanism in large IoT networks (Seyfollahi and Ghaffari 2020).

Introducing the load balancing mechanism in RPL will help to provide a unified connection in the network, and reduce latency, and power consumption to meet the IoT applications' requirements. In previous studies, the researchers were concerned about the poor performance of RPL in different networks. They proposed RPL based load balancing routing algorithms responsible for solving their network problems (Sebastian and Sivagurunathan 2018; Taghizadeh et al; Bobarshad and Elbiaze 2018; Altwassi et al 2019; Zarzoor 2020).

In various network settings, the proposed load-balancing routing algorithms behaved differently. For example, lbRPL according to Sebastian and Sivagurunathan (2018), improved network lifetime in Low power and Lossy network (LLNs). It proved to be reliable in a fixed network; its reliability was only evaluated on packet delivery ratio, control overhead, and node participation. Its latency and power consumption capabilities in IoT networks are unknown. The Context-aware and load-balancing RPL for IoT network (CLRPL) outperformed RPL in a heavy and dynamic IoT network (Taghizadeh et al 2018). The algorithm balanced the load considering the remaining queue, and the energy level of the parent node. It was evaluated in different traffic networks, and it improved the network's lifetime and stability. However, it is prone to control overhead due to the congestion notifications generated on the network. Among 30 nodes, the load balancing RPL for IoT in bust conditions outperformed RPL in terms of packet delivery ratio and power consumption for various network traffics. The method balanced the load by employing parent count via the ETX metric.

The existing RPL load balancing algorithms apply the load balancing in different ways and performed differently in different network traffic. The differences in the existing RPL load-balancing routing algorithms make it difficult to find effective, reliable, and stable RPL load-balancing routing algorithms for IoT network scenarios.

IoT applications require effective data transmission, and this study proposes triple RPL-based load-balancing routing algorithms with a common routing metric (buffer occupancy) for this purpose. The proposed RPL-based load balancing routing algorithms are known as Enhanced Context-Aware and Load balancing RPL(ECLRPL), Buffer occupancy load balancing (BLRPL), and Enhanced Expected transmission count RPL(EN-ETXPC-RPL). We implemented the algorithms in Contiki OS and demonstrated the IoT network in the Cooja network simulator to

observe the communication devices and performance of the proposed RPL-based load-balancing routing algorithms in different network scenarios. The algorithms' performance was assessed using stability and reliability metrics such as packet delivery ratio, power consumption, end-to-end delay, rt-metrics, and control overhead.

## 1.2 Problem Statement

The revolutionary Internet of Things (IoT) connects common objects to the internet. Various industries like healthcare, retail, manufacturing, and construction use IoT to operate effectively. IoT devices are inefficient and low-powered, with limited processing and storage capacities. These IoT devices generate large amounts of data that are challenging to handle. As a result, it is crucial to employ an effective routing protocol in this IoT network scenario.

The RPL IPv6 protocol was proposed by the Internet engineering task force working group as a solution for low-power and lossy networks. RPL is a promising protocol with multiple advantages for small devices; one of its strengths is its ability to adapt to and manage network topology changes and implementation modifications. In addition, it is utilized to address the needs of LLNs, such as load balancing, uneven energy usage, and network traffic. To ensure efficient data transmission in IoT networks, load balancing is RPL's most significant challenge.

Various researchers have tried to tackle this problem and proposed different RPL-based load-balancing routing algorithms. Some of the researchers are: (Sebastian and Sivagurunathan 2018; Altwassi et al 2019; Sutagundar et al 2019). The proposed RPL-based load balancing routing algorithms provide partial load balance which is limited to the network they deployed to, when they are applied on another network scenario they perform differ which means they are not optimal for other network scenarios. The advanced technology implemented in IoT applications requires optimal data transmission. Using the effective load-balancing routing algorithm can improve data transmission in IoT networks.

To find the effective RPL-based load-balancing routing algorithm, this study proposes three RPL-based load-balancing routing algorithms namely ECLRPL, BLRPL, and EN-EXPC-RPL, and evaluates their performance in IoT network scenarios.

## 1.3 Research Questions

This study addresses the following research questions.

- I. What is the state of the art of load balancing in IoT routing protocols?
- II. How can we identify effective RPL-based load-balancing routing algorithms for IoT networks?
- III. What recommendations can be drawn from the performance results emanating from RPL-based routing algorithms evaluation?

## 1.4 The Research Aim

This study aims to find an efficient RPL load-balancing routing algorithm for IoT networks. The effective RPL load-balancing algorithm may depend on the routing metrics and objective function. However, that would be achieved by investigating the performance of the proposed RPL-based load-balancing routing algorithms (ECLRPL, BLRPL, and EN-ETXPC-RPL) on the Internet of Things network.

The following objectives were formulated to achieve the goal of this study. They are arranged to systematically address the research questions raised in this study.

- i. To establish the state of the art of load balancing in IoT routing protocols.
- ii. To propose, design, and implement RPL-based load-balancing routing algorithms in the Contiki operating system.
- iii. To evaluate and compare the performance of RPL-based load-balancing routing protocols in IoT networks.

## 1.5 The rationale of the Study

The Internet of things (IoT) plays a crucial role in industries by enabling devices to collect and distribute data autonomously. The vital data collected and provided by the devices must be transmitted on time. In IoT networks, the routing protocol for low-power and lossy networks (RPL) is responsible for efficient data transmission. However, RPL does not adequately balance the load distribution in IoT networks, resulting in inadequate network data transmission. Several authors like (Sebastian and Sivagurunathan 2018; Taghizadeh et al 2018 and Abdullah et al 2021) have suggested the load-balancing routing algorithms RPL. The proposed load-balancing routing

algorithms performed differently based on the deployed network and provided partial load balancing in the network. Because of the performance differences, determining the most efficient RPL load-balancing routing algorithm for IoT networks is difficult. This study aims to find the effective RPL load-balancing routing algorithm for IoT networks by proposing three RPL-based load-balancing routing algorithms (ECLRPL, BLRPL, and EN-ETXPC-RPL) and evaluating their performance in the same IoT network scenario.

This dissertation will help to develop an effective RPL load-balancing routing algorithm for IoT networks since it identified buffer occupancy, remaining energy, and ETX as the routing metrics to be considered when designing the RPL load balancing for IoT scenarios. Since the current RPL load-balancing routing algorithms operate well in a variety of network settings, the study was done to find the best RPL load-balancing routing algorithm for IoT networks.

## 1.6 The contribution of the study

The IoT network is improving communication in different industries including e-learning, smart agriculture, home automation and healthcare. Wireless sensor devices play a vital role in sensing, monitoring, and tracking activities in IoT applications and therefore the communication in these applications must be reliable and timely.

Reliable data transmission in the network is made possible by the RPL standard routing protocol for low power and lossy networks, which is implemented in wireless devices. In IoT applications, RPL requires a reliable load-balancing approach to successfully transmit data in the network. This study explored the performance of the existing RPL-based load-balancing routing algorithms in different network scenarios to find the scalable and stable routing algorithm on the network. The explored RPL-based load-balancing routing algorithms use different routing metrics to stabilize the network. Different routing metrics were utilized to balance the load distribution results partial solution based on the network scenario they are deployed to. This brings confusion about which is a reliable and effective RPL load-balancing routing algorithm for IoT networks.

To fill the gap left by existing RPL load-balancing routing algorithms, this study proposed three RPL-based load-balancing routing algorithms for IoT networks which use three different routing metrics. The proposed RPL-based load-balancing routing algorithm consists of a single common routing metric and two others that are different. They are named as follows:

Enhanced context-aware and load-balancing routing algorithm for RPL (ECLRPL),

Buffer occupancy load balancing routing algorithm for RPL(BLRPL)

and Enhanced Expected transmission count and parent count load balancing for RPL (EN-ETXPC-RPL).

The proposed RPL-based load-balancing routing algorithms were implemented in Contiki Operating Systems in this study to discover the most efficient RPL-based load-balancing routing algorithm for IoT networks. The study also evaluated the performance of ECLRPL, BLRPL, and EN-ETXPC-RPL in IoT networks based on reliability and scalability performance metrics. Different network densities were used to test the performance in the Cooja network simulator. To ensure that, the proposed RPL-based load-balancing routing algorithms are effective and reliable for IoT networks, a performance comparison between the existing RPL-based load-balancing routing algorithms and the proposed RPL-based load-balancing routing algorithms has been conducted in the IoT network.

This study made contributions to the body of knowledge based on publications which are summarized below:

Initially, the study explored the existing RPL-based load-balancing routing algorithms in different network scenarios. This endeavor curated the different objectives utilized to balance the load distribution in the network as well as the effective objective functions and routing metrics for balancing the load distribution in RPL protocol. The publication is cited below.

1. Z. Magubane, P. Tarwireyi, A. M. Abu-Mahfouz and M. O. Adigun, "RPL-Based on Load Balancing Routing Objective Functions for IoTs in Distributed Networks," *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2019, pp. 1-6, doi: 10.1109/IMITEC45504.2019.9015879.

Based on the existing RPL-based load-balancing routing algorithms, objective functions, and routing metrics, RPL-based load-balancing routing algorithms: called ECLRPLL, BLRPL, and EN-ETXPC-RPL were proposed. They were then deployed in different network scenarios and their performance was evaluated based on different performance metrics to understand their reliability and stability in IoT networks. This aspect of the study was disseminated as follows:

2. Z. Magubane, P. Tarwireyi, A. M. Abu-Mahfouz and M. O. Adigun, "Performance Evaluation of ETXPC-RPL Routing Algorithm in IoT Network," *2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2020.
3. Z. Magubane, P. Tarwireyi, A. Abu-Mahfouz and M. Adigun, "Extended Context-Aware and Load Balancing Routing Protocol for Low power and Lossy Networks in IoT networks (ECLRPL)," *2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2021.
4. Z. Magubane, P. Tarwireyi, A. M. Abu-Mahfouz and M. O. Adigun, " Buffer Occupancy Load Balancing routing algorithm for RPL in IoT network (BLRPL)" **Accepted for publication** in the *Journal of Theoretical and Applied Information Technology (E-ISSN 1817-3195 / ISSN 1992-8645)*, 2023.

## 1.7 Organization of the dissertation

This dissertation is arranged into different sections.

**Chapter 1** Introduces the dissertation, outlines the objective of the study, which is to discover an effective RPL-based load-balancing routing algorithm for IoT networks, and describes the objectives to be attained and research issues of the study. It concludes by providing the dissertation's framework.

**Chapter 2** describes the background of the study which reveals the history of IoT and the operation of IoT in diverse application Scenarios, the importance of routing in IoT, and the Routing protocols for IoT applications. It also describes the standard protocol routing procedure and its drawback as well as the importance of balancing the load distribution in IoT applications.

**Chapter 3** contains the literature review of related work which starts by describing the importance of effective data transmission in IoT networks and later the existing RPL-based load-balancing routing algorithms and their performance in different network scenarios. The section highlights the tools utilized for performance evaluation and the drawbacks of the algorithms. The section is vital for this study because it shows that there is still a lot to be done to achieve effective data transmission in IoT networks since the existing RPL load-balancing routing algorithms focused on different network scenarios.

**Chapter 4** describes the methods used to conduct this study including operating systems used to implement routing algorithms, simulations parameters, and objective functions used for load balancing in IoT networks.

**Chapter 5** depicts the performance of the proposed RPL-based load-balancing routing algorithms in IoT networks. Performance is displayed in graphs based on various performance metrics that validate the reliability and stability of each routing algorithm in an IoT network. The chapter also analyses the performance of the routing algorithms in IoT networks.

**Chapter 6** concludes this study; it provides a summary of the entire study. It provides a prediction of future studies based on the results obtained in chapter 5. Finally, it gives recommendations on the future RPL load-balancing routing metrics for IoT networks that should be used by any scholar who wants to develop a new load-balancing routing algorithm for IoT networks.

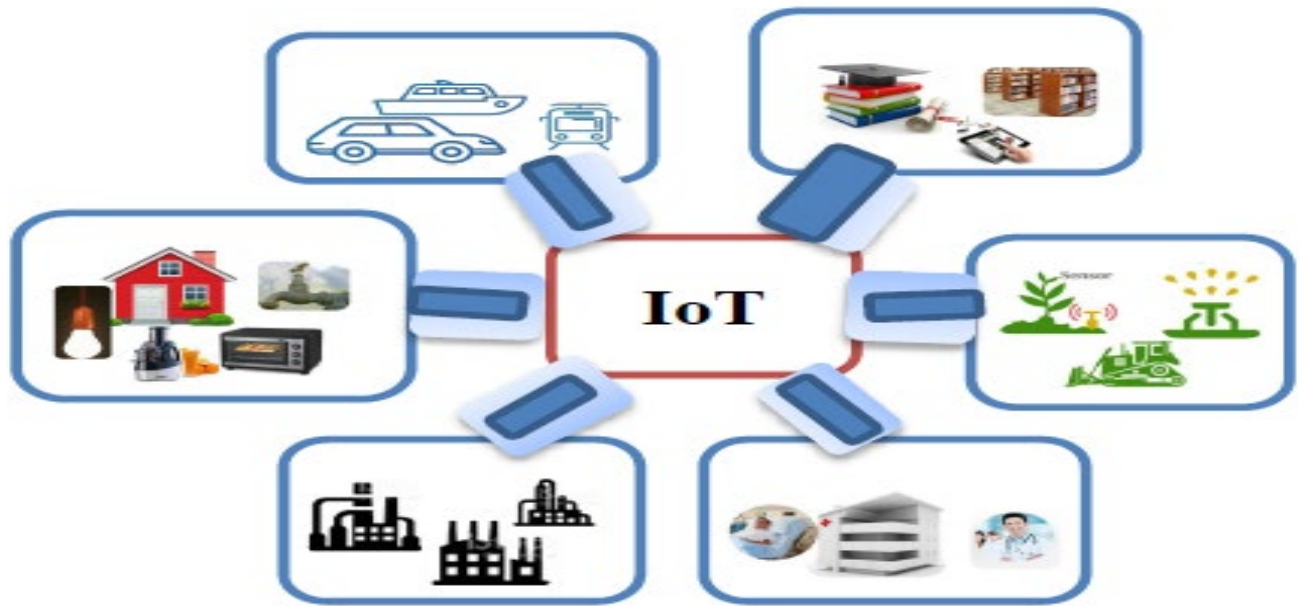
# Chapter 2

## Background of the study

This study is driven by several research questions and objectives that aim to find an effective RPL load-balancing routing algorithm for the IoT network. This section is intended to answer the first research question which is "What is the state of the art of IoT routing protocols" by providing a thorough review of IoT routing protocols. The chapter provides the background of the study and highlights the use of IoT in different industries and the deployment of IoT devices in different areas of the network. It also provides the operating procedures of the IoT routing protocol.

### 2.1 IoT

The IoT refers to the network of wireless and wired devices that link everyday objects to the web. It allows the gadgets to talk to one another via the internet and share vital data. The gadgets are embedded actuators and sensors that can communicate and make decisions in real time (Sankaran 2015). Memory, processing speed, battery life, and network capacity are all areas in which IoT devices suffer from severe limitations. IoT impacts socio-economic development such as smart homes, health, transportation, and industrial domains as shown in Fig.2.1. In healthcare, IoT introduces the simultaneous monitoring and reporting where the lives of the patients are effectively saved in the medical emergency event by monitoring the real-time issues via connected devices that share the information about heart failures, asthma, diabetes, clogged veins, etc.



*Figure 2. 1: IoT applications*

Health data such as oxygen, blood sugar levels, blood pressure, ECGs, and patient weights are collected and transmitted by the IoT devices. This information is securely saved in the cloud so that authorized parties such as your doctor, insurance company, health care provider, or outside consultant can access to it from any device, location, and at any time. IoT offers machine-to-machine connectivity, data mobility, and interoperability, all of which are essential to providing efficient healthcare services to patients. Data from IoT devices may now be gathered, published, and analyzed in real time, reducing the need to retain raw data. The collected data assists medical practitioners to know what is wrong with the patient and what medication will work best to treat patient. In the event of an emergency, IoT offers remote support that enables patients to obtain doctor information, such as location, specialization and contact details, which they can use to call a doctor who is miles away from them using sophisticated mobile apps. Quicker diagnosis of patients' conditions is made possible through mobility solutions. During the covid 19 pandemic, wearable devices were made possible by the Internet of Things and greatly improved the quality of life for hospitalized individuals. Pharmaceuticals are typically distributed through healthcare supply chains in response to prescriptions and patient health information collected through connected devices. As the IoT monitors patients, personnel, and inventory, it improves medicine management, guarantees the availability of essential hardware, and decreases the amount of time patients must wait in emergency rooms. This is a real-time application scenario where the process

should be fast and effective. Therefore, a reliable routing protocol for data transmission is highly demanded. IoT healthcare is cost-effective, highly secure, and less time-consuming (Al-shargabi et al. 2020). The Internet of Things introduces smart farming in the agriculture industry where the sensors collect the data and communicate the same to provide real-time monitoring. IoT keeps track of various factors such as temperature, soil texture, and humidity that save the farmers time and reduce the unnecessary use of water and electricity (Varshney et al. 2021). Adopting IoT in agriculture enabled farmers to have better information about climate change, precision farming, smart greenhouses, and drones to monitor the health of the crops. Monitoring soil moisture, temperature, and humidity improve productivity while being cost-effective (Kiani and Seyyedabbasi 2018). The communication of IoT devices is primarily affected by the distance between the devices and the load each device is carrying. When devices are closer to each other and share the data, it is faster than the far devices and the devices which communicate a large amount of data deplete their energy fast. However, the proper routing algorithm is needed for reliable data communication and security of data. The authors proposed cluster-based routing to protect the shared data and ensure high data aggregation. The routing algorithm ensured that all data was transmitted at a minimum cost and in real-time (Jain et al 2021). The algorithm promotes data transmission at a minimum cost of energy and with highly secured data to avoid a long time of service without battery depletion.

IoT enables smart transport which is an easy payment method where the integrated transport has a smart card loaded with money. The smart card can be swiped at any gate or terminal to get access to the transportation network. Near Field Communication (NFC) is the technology that transmits data from the card to the reader. This process conserves time for travelers by automating the payment even in parking lots, the car with tags is easily detected and the driver can know the parking charges as time goes on (Dlodlo et al. 2016).

IoT in the military industry automate the physical fitness test on the candidate during the recruitment process. Each candidate wears body sensors that verify the fitness of the candidate and send the obtained result to the doctor's file immediately. Adopting IoT devices in the military industry would provide the assurance of reliability and quantity in the battlefield environment. Training soldiers is conducted with a sensor that is usually embedded in individuals as a personal sensor system for health. The personal health sensor system collects the heart rate of the person,

fitness, and any related body issues to ensure that the person is ready for deployment. The data collected is used for deploying the right person in the environment (Fraga-Lamas et al. 2016). The IoT uses wearable devices to measure fatigue and detect problems by measuring the physical state of soldiers for the military application, mobility of nodes, securing data, and energy efficiency are the most important metrics. Different industries have sensors deployed to collect and communicate the information collected, due to IoT, sensor data can be transmitted online. The sensors generate different data traffic types in industrial applications and that traffic needs to be attended to immediately. Due to large traffic forwarded to the neighbor nodes, those neighbor nodes get congested and drop the packets, use too much power and eventually fail. The dropped packets contain sensitive information meant to make changes to human life. Routing in IoT devices helps to avoid congestion and improve the performance of data exchange. The congestion control framework has been proposed for industrial IoT to support the transmission of heterogeneous data, 6TiSCH network (Farag et al. 2020). IoT will make self-driving and autonomous cars possible, where the passengers can socialize, relax, and work in cars without driving them. Automated cars can reduce road accidents and lower energy consumption.

IoT automates things in different industries. The device responsible for data transmission has limited capacity, low battery, and limited processing speed. Various IoT applications encounter problems in the network where the nodes die early, resulting in the retransmission of a packet and loss of data. To optimize data transfer in IoT network settings, a reliable routing protocol is needed. In IoT networks, different routing protocols are available for transmitting data, and all routing algorithms operate at various layers, including the network layer, transport layer and application layer. All the routing protocols work together to improve performance; each protocol has its duty such as User Data Protocol (UDP) and Transmission Control Protocol (TCP) which in the transport layer are responsible coordination and transfer of data between end devices. The network layer has the following protocols COAP, RPL, and 6LoWPAN which are responsible for data routing between the devices. Depending on the needs of the network, the protocols found in the network layer perform variously. The network layer routing protocols are in charge of ensuring accurate operation of IoT applications by ensuring reliable data delivery in IoT devices.

## 2.2 Protocol stack

The protocol stack is a mechanism that enables the communication of IoT devices over the network. The large code and memory footprints of conventional TCP/IP implementations make them inappropriate for use with smart objects that are otherwise constrained. Rather than including all of the functionality found in a standard TCP/IP stack, the WSN TCP/IP stack is optimized to have only the most fundamental components. The four-tiered design of the WSN protocol stack is depicted in Fig 2.2.1.

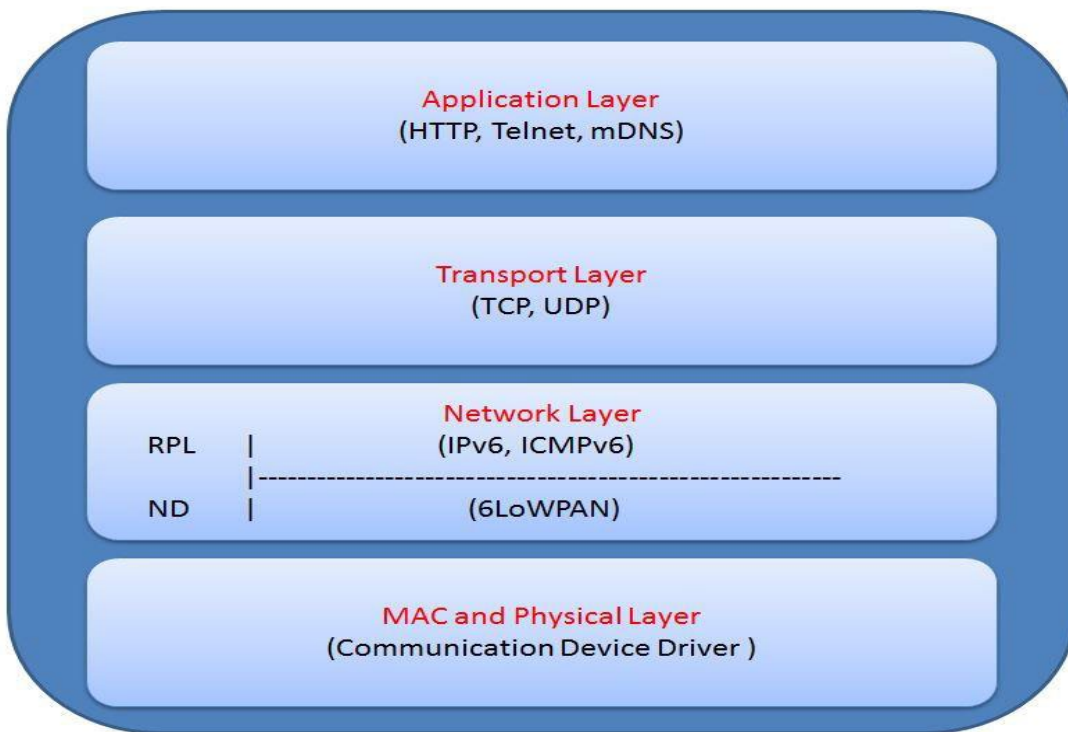


Figure 2.2. 1: Contiki protocol stack (Seliem 2011)

1<sup>st</sup> layer is the Physical and MAC layer.

The layer is responsible for physical device connectivity through a wired or wireless mechanism and the identification of the device by MAC address for physical addressing. Most standards combine the physical and MAC layer protocols to improve the establishment of the communication channel. MAC layer provides efficient bandwidth allocation, and power control mechanism and removes hidden and exposed problems in IoT devices because they are low-powered and have low processing capabilities. Power Line Communication (PLC), IEEE 802.11, and IEEE 802.15.4

communication protocols are used at the physical layer. This research will focus on IEEE 802.15.4 because it is widely deployed on such networks despite its low-power and lossy characteristics. IEEE 802.15.4 is the de facto standard radio technology for low-power networks but has a limited transmission range of only a few meters.

2<sup>nd</sup> layer is the Network layer.

Effective packet transport from source to destination nodes and logical addressing are the responsibilities of this layer. The layer is usually called a routing layer, at the network layer of the protocol stack, RPL is used to direct data packets to their destinations. The RPL protocol builds a network tree topology using control packets (DIO, DIS, and DAO) and maintains routing paths. If any of the node's unicast or multicast addresses do not match the packet's destination address, the layer sends the packet. Either because it is intended for it or that node, or both, the packet will be routed to another node. IPv6 is superior to IPv4 for use in WSN. The brief support for a simple header option in IPv6 provides the address necessary for IPv6 to properly enable operation in WSN. Cross-layer compressions are easier to achieve in an IP6 structure.

3<sup>rd</sup> layer is the Transport layer.

UDP and TCP are the protocols that make up this layer. The layer verifies the packet's integrity by computing a checksum, and then sends it to the appropriate application via the port number. It is the job of this layer to ensure that data is sent to the host successfully. It controls the congestion in the devices, improves data integrity, and controls the flow of data in IoT devices. This layer gives the application layer services that it receives from the network layer.

4<sup>th</sup> layer is the Application layer.

For any Internet of Things (IoT) device, the application layer is the point of contact between the device and the network. A client and server architecture forms the foundation of the application layer. It arranges the information in a manner that can be understood by both the client and the server. In order to access the services provided by the levels below, applications can utilize the API provided by this layer. It uses common communication methods including the Hyper Text Transfer Protocol (HTTP), the Telephone Internet Protocol (telnet), and others and is structured at the upper part of the framework. The application layer protocols are responsible for improving

reliability, effective bandwidth allocation, avoiding data latency, and transportation of data. It wants to converge the IoT social needs and industrial technology.

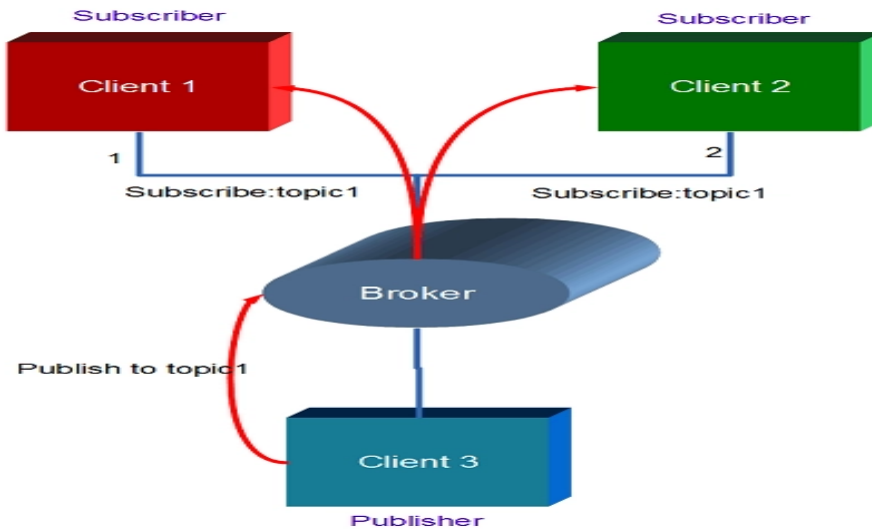
## 2.3 IoT Routing protocols

The Internet of Things connects all devices, allowing them to exchange the collected information from home (home devices), body(wearables), and industries (smart objects) without human interaction. The communication of these devices is challenging at some point because they are resource-constrained, with a limited capacity, battery power, and processing speed. The poor performance of IoT devices in different applications is addressed by routing data between the devices. Routing over low power and lossy networks (ROLL) developed a routing system for IoT device data transport. The following protocols determine their impact on IoT devices' communication.

### 2.3.1 MQTT

The Message Queuing Telemetry Transport (MQTT) is a lightweight application layer messaging protocol based on the publish/subscribe modes. MQTT is designed to improve communication of low-power devices, with limited bandwidth and high latency. It performs proper communication in Machine to Machine (M2M). The protocol provides communication solutions for the Wireless network by allowing the servers (brokers) to resend messages lost due to an unexpected disconnection. However, the communication is reliable but delayed(Chen et al 2020).

MQTT avoids direct communication of devices by providing a service which is an MQTT broker that enhances the communication between the devices. MQTT ensures that the data is delivered successfully to the required destination because it uses a connection-oriented TCP connection. The protocol supports machine-to-machine communication. The route selection is not a mandate that includes many steps, but the shortest path is preferred. The illustration of MQTT operation is depicted in fig.2.3.1.



*Figure 2.3 1: MQTT operational Diagram*

MQTT outperforms Hypertext Transfer protocol (HTTP) by reducing control overhead during the communication of data. MQTT is capable of reducing data transmitted by several devices by a quarter compared to the data transmitted by HTTP, the communication among the devices may not be secured (Sasaki and Yokotani 2019). When all the devices send their data to a single broker, it will take time to process and forward that data. However, some IoT devices transmit sensitive real-time data that needs to reach the destination as soon as possible and not be changed, MQTT protocol may not be a reliable protocol because it delays the data transmission and does not consider the security of the data.

### 2.3.2 CoAP

An application layer called Constrained Application Protocol (CoAP) serves as a simplified version of Hypertext Transfer Protocol (HTTP). CoAP is intended to be usable with sensor devices that lack resources. User Datagram Protocol (UDP) is a preferred transport protocol to transmit request and response messages in CoAP because it offers lightweight services. CoAP is commonly used in an application that is controlled remotely like a smart home application (Thirupathi et al. 2018). Some of the nodes in CoAP act as clients that are responsible for sending the data to the server node. Quality of service is a key component of the protocol, since it is used to regulate communications by marking messages as confirmable or non-confirmable to signal whether or not the recipient should respond with an acknowledgment. It helps with the method for negotiating content and finding resources, as well as setting up encrypted communication.

### 2.3.3 CARP

Distributed routing protocols created specifically for underwater communication include Channel-Aware Routing Protocols (CARP). Due to its small packet size, it can be used for IoT. Link quality, which is calculated based on historically successful data transfer gathered from nearby sensors, is taken into account while choosing forwarding nodes. The two cases are data forwarding and network initialization. A HELLO packet is broadcast to every other node in the network during in-network startup from the sink. Data forwarding involves sending a packet hop-by-hop from the sensor to the sink. Each hop after that is decided on its own. The primary issue with CARP is that it does not encourage the reuse of previously gathered data. In other words, if an application only needs sensor data when something drastically changes, CARP data forwarding is not advantageous for that particular application. By enabling the sink node to store previously received sensory data, E-CARP improved CARP. E-CARP dramatically minimizes communication overhead by sending a ping packet when data is required, which is then answered with the data from the sensor nodes (Basagni et al 2015).

### 2.3.4 LOADng

One of the most important protocols for effective routing in Low power and Lossy networks is the Lightweight On-Demand Ad-hoc Distance Vector Routing Protocol-Next Generation (LOADng). It was created specifically for the Mobile Ad-hock Network (MANET). LOADng is an alternative to cover up the RPL drawbacks in Mobile devices, it enables the nodes to transmit packets based on their destination of choice. Hop count, ETX, Link Quality Indicator, and Minimum Battery Cost are only few of the routing metrics that the LOADng uses to complete a routing task. These routing metrics are used to determine the best path from a source node to an eventual destination (Sobral et al. 2017).

The protocol transmits packets by generating a request and waiting for a reply message before the data is transmitted. If the destination and route are unknown, the node must keep the packets in a queue and initiate a route discovery method before sending them to the destination. A node sends a Route Request (RREQ) to discover a destination, then the node that receives the RREQ makes an entry on the Routing set for the message originator and verifies if it is the destination before deleting or rebroadcasting the packet RREQ. When the same message is already received or when the hop restriction expires, a packet is dropped. When a node successfully reaches its target, the

destination node broadcasts the Route Reply (RREP) message to provide the address based on the same path traveled by the RREQ message (Mehta and Rekha 2020). As indicated in fig.2.3.4.1, the RREP message must reach the node that broadcast the RREQ before the packets are unicast from the source to the destination node.

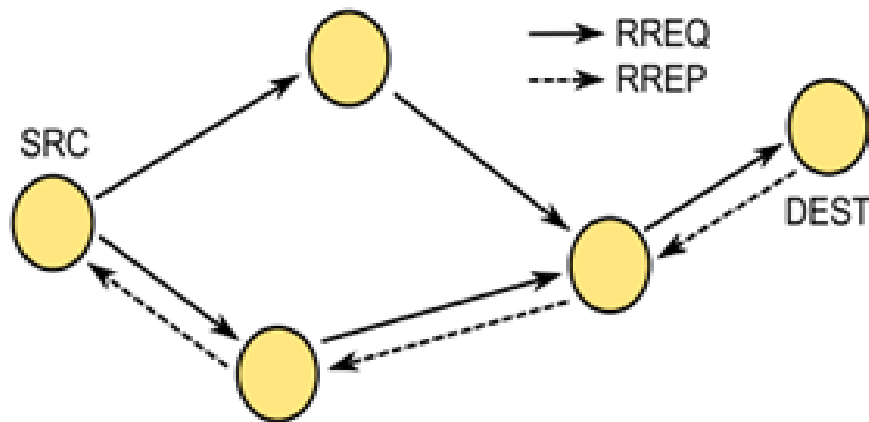


Figure 2.3.4.1:LOADng protocol operation (Sobral et al. 2019)

As the need for network connectivity grows, the authors propose an enhanced method for data delivery on mobile IoT devices using the LOADng protocol (Sobral et al. 2019). As a result of the route discovery process, the LOADng enhancements raised the control overhead, network delays and incur significant power consumption. While RPL performs better than LOADng protocol in Multi-Point to Point (MP2P) connection, LOADng is more dependable for point-to-point(P2P) communication (Clausen et al. 2019). For these reasons, LOADng is unreliable for IoT applications compared to the RPL protocol.

### 2.3.5 RPL Overview

Some networks, known as Low-Power and Lossy Networks (LLNs), have limitations on their routers and intermediary nodes. Instability in the network, high loss rates, and low data speeds are all results of the limited memory and processing capability of LLN routers. Therefore, the Routing Over Low power and Lossy Networks (ROLL) working group designed an IPv6 Routing Protocol for Low power and Lossy network (RPL)(Winter et al 2012), which was standardized by the Internet Engineering Task Force (IETF) as a standard routing protocol LLNs and documented it on RFC 6550. This protocol uses a subset of ICMP called RPL Internet Control message Protocols to communicate information between nodes based on their relative positions and directions in the

network topology tree. The RPL protocol organizes the network with Destination Oriented Directed Acyclic Graph (DODAG) which forms a tree-like routing topology. The nodes join the RPL network using control messages known as DODAG Information Solicitation (DIS), DODAG Information Object (DIO), and Destination Advertisement Object (DAO).

The RPL protocol has the trickle algorithm which set time intervals for sending the control messages (Levis et al. 2011). The time intervals are responsible for minimizing the control messages, DODAG is periodically refreshed with the trickle algorithm to stabilize the network and advertise the recently received data (Goyal and Chand 2018).

A variety of traffic types are supported by the protocol, such as Multi-Point-to-Point (data transmission from LLN nodes to the root node), Point-to-Multi-Point (data transmission from the root node to the LLN nodes), and Point-to-Point (data transmission among the nodes inside the LLN). RPL has a set of routing attributes that are regarded as routing metrics or routing constraints. The routing metrics are responsible for determining the reliable links and nodes for effective data transmission from the candidate node to the root node (Winter et al 2012).

RPL has objective functions to guide the functioning of the routing metrics and constraints to reach a specific goal. The routing of packets in RPL is optimized by objective functions using ICMPv6 control messages by creating Destination Oriented Directed Acyclic Graph routing paths (DODAG)(Parsaei et al 2018).

## **RPL topology formation**

The RPL standard specifies the control messages necessary to set up and maintain the network architecture. DODAG is a control tree topology formed by three ICMPv6 messages: DODAG Information Solicitation (DIS), DODAG Information Object (DIO), and, optionally, Destination Advertisement Object with Acknowledgement (DAO+Acknowledgement) (DAO-ACK). Occasionally, the receiver will utilize the message to confirm receipt of the packet from the sink node.(Onwuegbuzie et al 2020).

- DODAG Information Object (DIO)

Root node generates DIO control message during network construction; it includes node rank, node id, RPLInstance, and node lifetime; DAG Metric Container stores link and node metrics or constraints. The information contained by this control message is needed by the

client nodes to find the effective routing path, and preferred parent to the root nodes. In order to keep the DODAG operational for the duration of the network's existence, the DIO messages are continuously broadcast via multicast to all sensors.

- Destination Advertisement Object (DAO)

In response to the DIO control message, the client nodes announce their network membership via a DAO message. Within a DODAG, the DAO messages often act as the upward link's propagator. A live routing table can be maintained by having child nodes send DAO messages to their parents along the path to the root.

- DODAG Information Solicitation (DIS)

The message type is used by the nodes that are not yet joined the network and seek DIO messages from RPL neighbor nodes. When neighbor nodes receive the DIO message, they will multicast DIO to neighbor nodes. When nodes receive DIO messages they can join this RPL network(Kalyani and Vydeki 2018).

- Destination Advertisement Object Acknowledgement (DAO-ACK)

The client node may utilize the control messages to confirm receipt of the DAO message from the root node. DAO-ACK verifies that the node has received the destination advertisement; if the node has not received the advertisement, it will not send an ack to the root node. Before any data packets are transmitted between nodes, the RPL network topology is shown in fig.2.4.5.1.

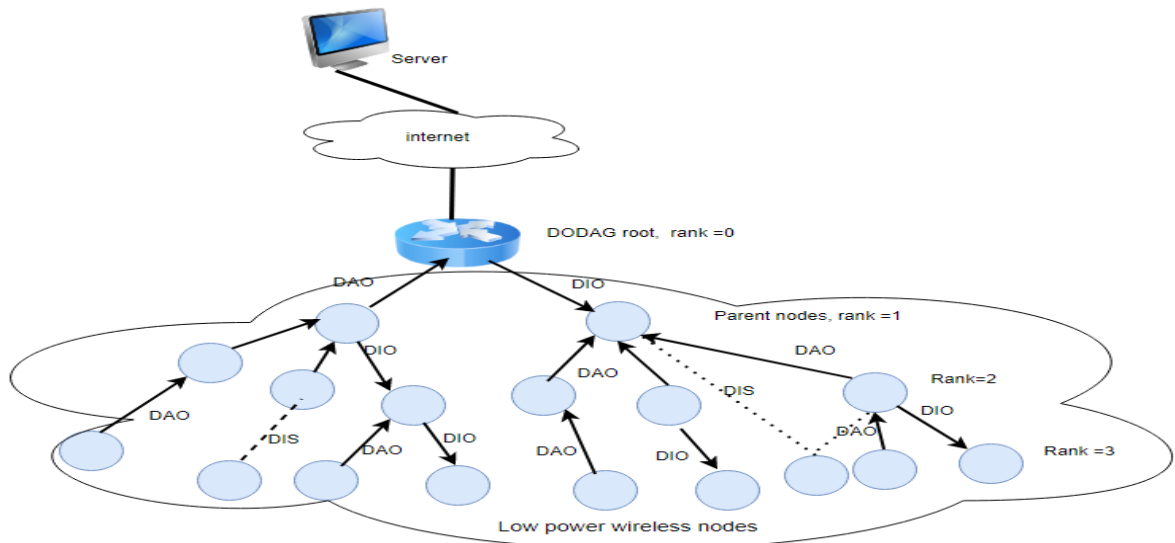


Figure 2.3.5. 1: Basic RPL operation with single DODAG in a single instance

## Objective functions

The RPL protocol applies the specific objective function to transmit data in various types of networks. In order to choose and optimize the routes inside an RPL network depending on the information objects provided, an RPL node must employ an objective function (Yassien et al 2019). The main idea behind the objective function is how various routing metrics will be combined and taken into account when constructing the route (Kuwelkar and Virani 2021). The routing metrics have quantitative values used to measure the routing path. However, the metrics can either be node metrics (the numerical values of the node's attributes) or link metrics (the optimal link between the nodes, assessed by link quality). Objective Function Zero (OF0) and Minimum Rank with Hysteresis Objective Function (MRHOF) are the two (2) objective functions that come standard with the RPL protocol (Thubert, 2012). (Gnawali and Levis 2012). The objective functions are applied separately to perform in different network scenarios. Due to differences in the routing metrics used, their operations are distinct.

### Objective Function Zero (OF0)

The default objective function zero makes sure that it looks for the closest grounded root. Based on the hop count routing measure, the objective function links the network's collection of large nodes and permits transmission of data from the source node to the root node. The child nodes can find the reliable parent node by looking for a parent node that is closest to the root. If there are no such parent nodes, the LLNs reform the floating DAGs that are rooted at the nodes with the highest administrative preference. By computing the rank increase of each node in the network, the OF0 chooses the node's path. The rank calculation is described by the equation (2.3.5.1):

$$Rank_{node} = Rank_{parent} + Rank_{increase} \quad (2.3.5.1)$$

Preferred parent ranking is defined by Rank parent, and preferred parent ranking increases are defined by rank increase, as shown in the equation (2.3.5.2) below:

$$Rank_{increase} = MinHopRankIncrease \times Step \quad (2.3.5.2)$$

MinHopRankIncrease is used to establish the scale of Rank increase, with the step recognized as a scalar number that indicates the quality of the rank increment along a path in the DODAG.

Without doing any sort of load balancing, OF0 chooses the preferred parent and alternate parent (if available) to send the packets to the root node.

## MRHOF

Objective functions for RPL can be provided, and one such function, the Minimum Rank with Hysteresis Objective Function, uses hysteresis to select the path with the lowest metric value. When sending data in RPL, MRHOF uses the metric determined by the DIO metric container. If the DIO metric container is empty, MRHOF will use ETX as a fallback metric to determine order.

MRHOF transmits packets using the standard link metric, Expected transmission count (ETX). The expected transmission count (ETX) is the predicted number of successful transmissions from a node. To ensure that all data packets reach the root node, each node in the network must use a secure connection to transmit them. The parent's ETX metric is determined by equation 2.3.5.3 below:

$$ETX = \frac{1}{D_f \times D_r} \quad (2.3.5.3)$$

$D_f$  is the chance that a packet is successfully received at a receiver node, and  $D_r$  is the likelihood that an acknowledgment message for that packet will be received. The equation (2.3.5.4) calculates the node's rank:

$$Rank_{node} = Rank_{parent} \quad (2.3.5.4)$$

ETX is the expected transmission count to the parent node, and Rank parent is the method by which that rank is calculated. When sending data packets across a network, MRHOF aims to reduce the associated route costs as much as possible.

### 2.3.5.1 Trickle Algorithm

The trickle timer is the algorithm that adjusts the sending rate of the control messages to improve network stability and energy consumption. It also controls the traffic during routing to set RPL protocol-free from routing loops. The trickle in the RPL protocol is used to advertise the most recently received data. It detects the inconsistency which is usually caused by the routing loops, new nodes joining the network, and the changes in the rank of the node (Clausen et al. 2011). The trickle communication grants the flexibility of operating in various network scenarios. However,

in the sparser network, the algorithm requires more transmissions per node while balancing the load.

The trickle algorithm consists of three (3) constraints namely:

1.  $I_{min}$  (minimum interval size) which is defined in units of time (milliseconds or seconds),
2.  $I_{max}$  (it is the value that doubles the  $I_{min}$  value).
3. redundancy constant  $k$  value. The algorithm has variables such as  $I$ ,  $t$ , and  $c$ . to determine the present interval size. The present time in the interval range is defined by  $t$ , while the counter variable is represented by  $c$ .

### **The trickle algorithm operation.**

The trickle algorithm is responsible for balancing the load distribution in IoT routing protocols by avoiding control overhead and mitigating power consumption. It enables the nodes to broadcast their configuration parameters at runtime. In the network, it is mostly used to advertise the most recent data received through a version number to avoid network overhead. When a node hears new data its update its data. When a node hears old data while it has new data, it reset itself and sends the updated data to all other nodes to avoid receiving old data and inconsistency in the network. As with any timer, a trickle's can be reset by anything happening in the outside world. The RPL protocol's trickle algorithm's operation is briefly outlined in fig.2.5.3.1.

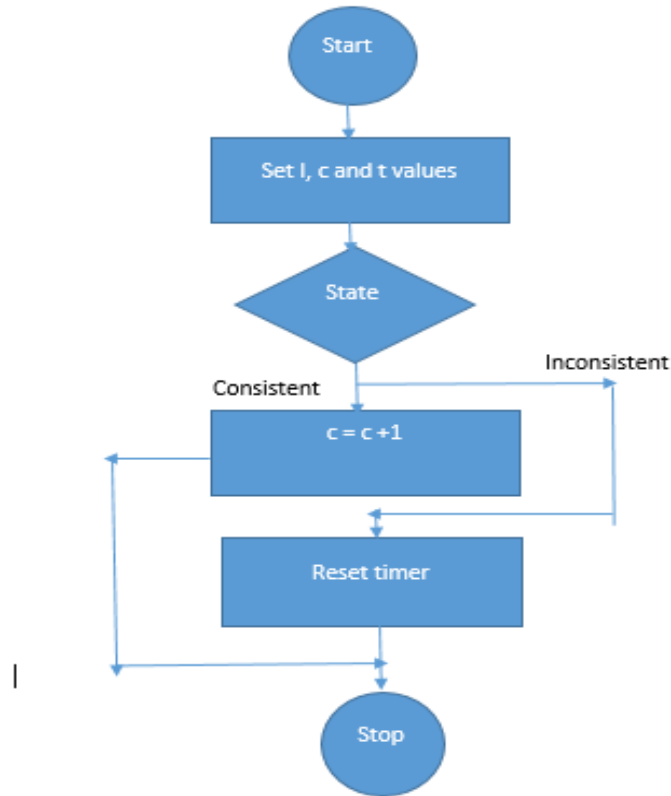


Figure 2.5.3.1: Trickle operation (*Trickle library – Contiki 2014*).

### 2.3.5.2 RPL routing metrics and constraints

The RPL uses an objective function and active routing metrics to route data using a distance-vector routing protocol. Metrics for determining the most trustworthy path between nodes may be based on links or on individual nodes themselves; this is known as "routing (Badach 2018). The designed routing metrics for the RPL protocol are explained below:

- Node state and attribute (NSA): the metric reveals the details about the characteristics of the node.
- Node Energy: provides information on the node's energy, including how much energy it had when it was first powered on and how much it has left after data reception and transmission.
- Hop-Count: indicates how many intermediate nodes must be traversed before reaching the parent node.
- Throughput: This shows the outcomes of compromising power usage for data transfer rate. As a result, the nodes advertise the maximum throughput their links can support.

- Latency: the constraint that reports the latency of the node
- Link Reliability: this metric can be measured by:
  - Link Quality Indicator (LQI) evaluates the link's dependability using a discrete number between 0 and 7..
  - Expected Transmission Count (ETX): count how many attempts a node thinks it will need to make to reach its destination before a packet is considered delivered.
  - Link Colour (LC): A 10-bit link is used as the constraint to define the connections between the various forms of communication.

Using the DAG metric Object Container object, these metrics are conveyed in the DIO message optional field. The flags are set up to show whether an item is metric or constraint, additive or not, and whether it is local or global. (Alayed et al. 2018). Both mentioned objective functions of RPL do not balance the load distribution during routing which makes RPL underperform in large IoT network scenarios where the data is heterogenous and transmitted in different rates. RPL has been standardized for LLNs among other routing protocols because it meets the requirements for wireless sensor networks. However, RPL does not balance the load distribution in the network. The section covered the routing protocols in IoT and specify their operation and effectiveness in IoT applications. The significance of balancing the load distribution while routing in an IoT network is discussed in the section below.

## 2.4 Load balancing

In IoT, the devices that collect data in IoT applications are deployed randomly in different areas; some of the nodes tend to carry heavy workload than other nodes. The nodes with high data become hot spots and consume more energy to process and transmit all the data, that leads to a network delay and high packet loss occurs. However, considering balancing load distribution among IoT devices can play a critical role. The goal of load balancing is to ensure that all nodes in a network receive an equivalent share of the data being sent and received, which will extend the life of the network, reduce power consumption, and reduce any delays in the data being sent or received. Improving network communication among IoT devices requires a complex load-balancing routing algorithm to be designed within a standardized protocol. The standard routing protocol for IoT, RPL protocol objective functions perform data distribution based on ETX and Hop count routing metrics that ignore the load balancing since RPL was designated to handle low data traffic

(Altwassi et al. 2019). In high-traffic scenarios or large networks, RPL performs poorly due to improper load balancing. Improper load distribution in high network scenarios may cause loss of data packets along the path and high congestion in the parent nodes. When the nodes are congested, they are regarded as the hotspot nodes, they consume more power to process data and the delay becomes higher. Multiple analyses were performed and analyzed in various network settings to determine the optimal load distribution for the RPL protocol. The proposed RPL-based load balancing routing metrics functions with different multiple routing metrics, multiple paths, or different clusters to improve network performance. Detailed information about the previously proposed load-balancing routing algorithms is contained in chapter 3.

IoT experts believe that a multi-path load-balancing algorithm will provide the best possible service (Sutagundar et al. 2019), however, it has a problem of duplication of data and frequent parent changes. Multiple paths improve the network reliability of the nodes by reducing load, which mitigates the energy consumption and bandwidth of the node.

RPL routing protocol has objective functions that provide better path selection based on the routing metrics. They compute the rank based on Hop count and ETX routing metrics, in large network traffic of IoT, RPL unfairly distributes the traffic among the nodes. Unfair data distribution on the nodes leads to network congestion, early depletion of energy on the congested nodes, and poor network performance. Many researchers designed RPL-based load-balancing routing algorithms networks of their choice. The algorithms seem to provide a partial solution in the network. Those partial solutions have limitations based on network size and performance parameters, hence RPL requires an efficient load-balancing routing algorithm for IoT networks.

## 2.5 Summary

In conclusion, IoT is the network that promotes automation and effective communication in different industries. The communication is driven by embedded sensor devices and actuators. However, the sensor devices have limited resources such as power, processing speed, and capacity. Due to the constrained nature of the connected devices, efficient routing is essential in the IoT. It has been shown that many of the data routing protocols introduced thus far are well-suited to low-power and lossy networks.

The Internet engineering task force group proposed the routing protocol for low-power and lossy network (RPL). The protocol transmit data effectively in low network traffic with its objective functions and routing metrics. The protocol has multiple objective functions that are unable to work simultaneously and a single objective function suitable for that network scenario must be chosen for each study. The objective functions transmit data using routing metrics; it can use a single metric or multiple metrics depending on the network scenario. Multiple routing metrics are mostly used by the modified routing algorithm for RPL where they are used to balance the load distribution and improve the original RPL performance. Depending on the control traffic messages, nodes choose a parent from which to send data, with the sink node acting as the network's DODAG root. The routing procedure for RPL raises an unbalanced load in high network scenarios. The imbalanced load issue in RPL became a research topic by many authors and several load-balancing routing algorithms were proposed. Most of the algorithms balance the load for a certain network scenario and outperform in different network performance metrics.

Here, we compared RPL's operation to that of other routing protocols. Many industries rely on IoT to perform their duties effectively. IoT devices generate heterogeneous data which is not easily transmitted by the standard protocol (RPL) because it does not balance the load distribution effectively. The scholars experienced the poor load distribution of RPL in different network scenarios and began to design the load balancing routing algorithms based on RPL. The following section describes the existing RPL-based load-balancing routing algorithms and the problems associated with the existing RPL-based load-balancing routing algorithms.

# Chapter 3

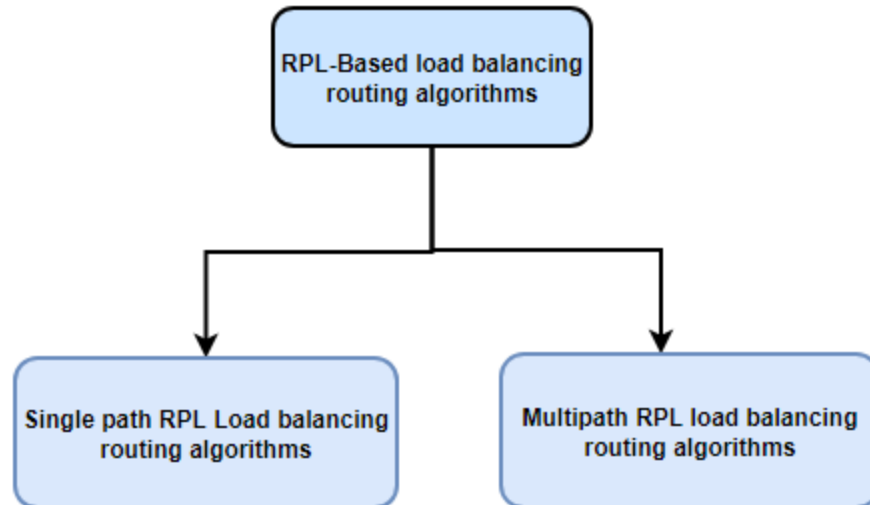
## A Review of RPL-Based Load Balancing approaches

### 3.1 Overview

A literature review is the research method of interpreting the existing related work of the study (Randolph 2009). It is utilized to establish the state of the art of the research. This study aims to find an effective RPL-based load-balancing routing algorithm for IoT networks. An analysis of the existing literature on RPL-based load-balancing routing algorithms can reveal important details about the algorithms' designs and the top-performing algorithms. This technique is used to identify the areas where the current load-balancing routing algorithms fall short, and it paves the way for the evaluation of the most efficient RPL-based load-balancing routing algorithms. This chapter explores the existing RPL-based load-balancing routing algorithms intending to find the effective load-balancing routing metric for RPL protocol.

The IoT applications generate heterogeneous data that have various requirements to be transmitted. Therefore, the routing protocol that is responsible for data communication in IoT devices encounters the problem of balancing the traffic distribution among the nodes to improve network lifetime. The load balancing problem in the RPL protocol has been studied in extensive works and in different approaches like optimizing the path selection, parent selection, child node count, and congestion awareness (Latib et al. 2017; Sebastian and Sivagurunathan 2018; Bhandari et al. 2018; Altwassi et al. 2019; Nguyen and Kim 2019). The authors discuss their proposed solutions to the problem of uneven load distribution in the RPL protocol and put those solutions to the test across a variety of network configurations. Many ideas were presented. Some of these include balancing the load by reducing network congestion, optimizing energy consumption, and optimizing network delay to improve network reliability.

The graphic in fig.3.1 shows the various categories of methods suggested to get around the RPL protocol's load balancing problem.



*Figure 3.1: Different categories of RPL-based load balancing approaches*

This chapter presents the investigation of the related studies which identify the RPL's current load balancing routing methods under various network configurations, network sizes, and tools used for evaluating its performance. This section aims to review the current literature and find the gaps that will lead to an efficient solution. The chapter is divided into sub-sections. Section 3.1 introduces the chapter, and gives a brief overview of the entire chapter, while Section 3.2 provides the detailed single-path RPL load balancing approaches. The third section (3.3) presents the RPL load balancing approaches based on multipath approaches, where the protocol utilizes multiple routes to reach the parent node to improve network lifetime in RPL protocol and the last section 3.4 presents the found gaps and opportunities in the related work.

## 3.2 Single Path RPL-based load balancing Routing Algorithms

The child nodes in RPL transmits packets to the root node through a single path. In RPL protocol, the root node is referred to as a border router since it operates as a gateway by transmitting LLN data to the internet. In a large network, a single gateway may receive packets from a single side of the network because of delays on the other side of the network. However, balancing load in single paths routing algorithms may improve the performance of information exchange by enabling all packets to reach the destination relying on the original design of RPL. This section examines the existing single-path RPL load-balancing routing algorithms.

Many sensors are used in IoT applications, and this results in a lot of data being transmitted and

received. When multiple devices send data simultaneously, the parent node that receives that data get congested and loses some packets. The studies were conducted to find ways to avoid congestion in the network caused by imbalanced traffic distribution, poor link quality, and or poor path selection. ETX routing metric provides the best link, but it does not check the available capacity and the remaining battery of the parent nodes to accommodate arriving data packets. This result in congestion, excessive packet loss, and short network lifetime.

Kim in (Kim 2015) proposed the actual queue utilization grounded on an RPL procedure named Queue Utilization-RPL (QU-RPL). It was intended that each node would be able to select its own parent node depending on its position in the neighboring node queue and its hop distance from the LLN border router. The QU-RPL algorithm was equated with the default RPL in respect to control overhead, hop distance, queue loss ratio and packet delivery ratio. The results demonstrated that QU-RPL enhanced the packet delivery ratio, reduced queue losses, and decreased end-to-end delay in Wireless Sensor Networks (WSN). QU-RPL excluded the ETX and parent count for the nodes to select the preferred parent which might lead to poor link quality selection. The queue utilization parameter is insufficient for load distribution balancing in RPL for IoT networks.

Al-Kashoash et al. (2016) proposed a Congestion Aware Objective Function (CA-OF) for RPL protocol. The research centered on buffer occupancy (BO) to identify crowded nodes and ETX to route data packets from the source node via the least crowded path back to the parents. The 2.7 version of the Contiki operating system, the Cooja simulator, and a testbed consisting of 19 and 35 nodes in a 6LoWPAN network were used to implement, analyze, and deploy CA-OF. Throughput, packet delivery ratio, and energy consumption were utilized to evaluate the CA-OF's performance. The objective function outperformed ENERGY-OF, OF0, and EXT-OF in terms of packet delivery ratio, energy consumption, and throughput although it has route instability. CA-OF mitigates congestion in the 6LoWPAN network by balancing the traffic distribution in child nodes. The focus of this algorithm was node metric to prevent buffer overflow.

Internet of Things devices are resource-constrained so they need more energy to deliver packets to their destination. In high-traffic networks, load imbalance and energy usage are challenges. Nodes gather data and transmit it to the sink node for processing, some of the nodes have inefficient

power so they send the packet and die during the process leading to high packet loss. Some scholars developed algorithms that balance the load with the purpose to optimize the energy consumption in IoT devices. Goyal and Chand (2018) proposed a protocol to improve the trickle algorithm that is useful for the consistency of the network in the RPL protocol. The proposed protocol balanced the load by providing a fair broadcast mechanism of the trickle-F algorithm.

The researcher proposed the Load Balancing Objective Function (LB-OF) in (Qasem 2018). This objective function was developed as an addition to the existing RPL objective functions. LB-OF was responsible for mitigating the packet loss and occurrence of power depletion in the burst network. The OF ensured that each parent node had an equal number of children. The effectiveness of the algorithm was measured by looking at how much energy it used to run, how quickly the first node died, and how well it delivered packets. LB-OF provided a better packet delivery ratio and consumed less energy. The algorithm still needs to be evaluated based on the link quality and network delay in RPL-distributed networks. Optimizing energy consumption should be not the only priority to balance the load distribution in RPL for IoT networks.

The heterogeneity of data transmitted in IoT applications drains the battery in IoT devices, therefore the researchers (Sennan, 2018) proposed an Energy and Load aware Routing Protocol (EL-RPL) for the Internet of Things. EL-RPL is a composite metric that enhances load distribution in an IoT network utilizing the optimal parent selection method. Routing variables including battery depletion index (DBI), load, and expected transmission count (ETX) are used to determine the optimal parent to assign a child to. EL-RPL was evaluated in the Cooja simulator in comparison with OF-FLRPL and RER(BDI)RPL objective functions. EL-RPL reduced parent changes during data transmission leading to less energy consumption high packet delivery ratio. EL-RPL has not been evaluated in a network larger than 30 nodes and a testbed scenario.

(Taghizadeh, Bobarshad and Elbiaze, 2018) proposed a routing protocol called Context-aware and Load Balancing RPL (CLRPL) to balance the load in heavy and dynamic networks. The algorithm had the aim of improving network lifetime in IoT-heavy and dynamic networks by reducing the rate of packets being lost in high throughput and highly variable traffic. The algorithm categorized its approach into sections to solve different problems sequentially. The initial part of CL-RPL is the newly composed objective function Context-Aware Objective Function (CAOF). Because of

CAOF, the parent and child nodes' ETX and residual energy could be used to determine the nodes' rank. The information in a DIO message is stored in an array on the node that received it. CAOF avoided the thundering herd phenomenon and optimized the parent selection using Context-Aware Routing Metric (CARF). In order to determine the value used in parent selection, CARF looks at the queue chain state along the path, the index of network traffic volatility, and the node's rank (calculated by CAOF). In addition, CLRPL kept the parent selection going by always going with the parent who had the fewest kids, as counted by CARF. The algorithm was effective in various traffic of 50 nodes network density, and it was limited only to the simulation environment. The congestion avoidance notification may result in high power consumption and control overhead.

Parsaei et al. (2018) proposed an RPL Load Balancing Protocol named LB-RPL for Ipv6 over Low Power Wireless Personal Area Networks (6LoWPAN) in order to achieve a more equitable distribution of heavy traffic. The LB-RPL protocol took network load differences into account and dispersed data flow across many parent nodes. The protocol altered the RPL control message creation procedure DODAG to prevent the node from delivering a fresh DIO packet immediately. The node initiated a timer, compared it to its workload, and, once the clock ran out, sent out the DIO packet. To indicate the workload, they introduced the parameter named buffer utilization counter. The parameter was specified as the average number of packets being pushed into the buffer. Three hours were spent simulating a network with two hundred and fifty (250) nodes to test how well the LB-RPL would perform. The simulator used was called Contiki Cooja. As a result, LB-RPL achieved higher throughput and a higher packet delivery ratio than RPL. The protocol took slower to send data packets to the root node and used other routes whenever traffic was heavy. The protocol prevented congestion, but it delayed the network performance that is a problem in IoT network because it must provide the realtime communication.

Balancing the traffic load in RPL protocol considering the energy consumption of the nodes is important because power has a significant impact on the network lifetime. The fuzzy logic algorithm is applied in RPL routing metrics to optimize path selection and parent selection. (Sankar and Srinivasan 2018) proposed Fuzzy logic Based Energy Aware Routing Protocol for the Internet of Things (FLEA-RPL). FLEA-RPL implemented fuzzification and defuzzification methods in Load distribution, ETX, and Residual Energy (RER) metrics. An updated RPL protocol has been implemented on the Contiki operating system. The research's major aim was to determine the

optimal parent through which packets might be forwarded to the DODAG root. To reduce energy use and network latency, FLEA-RPL balanced the load by counting the number of children per parent node. FLEA-RPL reduced network delay and network lifetime while increasing the packet delivery ratio. The study did not take into account the amount of electricity used by every node in the network.

Sebastian et al. (2018) designed a load-balanced routing metric for RPL (lbRPL). The lbRPL routing metric was developed to optimize load balancing. The lbRPL worked with the formulation of the LBI (Load Balancing Index) and parent selection for load balancing parent count (PC). LBI took advantage of the load-balancing capabilities of RPL nodes to choose load-balanced routes to the parent node. The functionality of LBI was evaluated based on network performance and network stability to increase the network's lifetime in the RPL protocol. Contiki OS and Cooja network simulator were used to evaluate LBI in comparison to ETX and parent count based on network lifetime, packet delivery ratio and control traffic overhead. The obtained results depicted that lbRPL performed better in network lifetime through node participation compared to ETX and hop count. In contrast, ETX performed better than hop count and lbRPL in the basis of control traffic overhead. The packet delivery ratio for ETX was 96.74%, hop count 95%, and lbRPL 96%. The results based on the packet delivery ratio showed that ETX performed better than other routing metrics used. The proposed method provided better network stability by choosing the load-balanced parents, but it still decreased the packet reception ratio. A low packet reception ratio has a bad influence on the network performance; hence, the load balancing algorithm is expected to enhance the network performance.

Enhanced RPL (EN-RPL) is an extension of the default RPL protocol proposed by (Singh and Chen 2019) for optimum load distribution in heavy and highly dynamic IoT networks. EN introduced its Efficient Routing Objective Function (ER-OF) that utilizes the coefficient routing metric to balance the load in RPL. In order to evenly disperse data in a busy IoT network, ER-OF allows child nodes to choose those with less demand. The technique was built in Contiki OS and assessed using the Cooja simulator. Compared to the original RPL, it enhanced the packet delivery rate and reduced energy usage. Balancing the load distribution has a an impact on the Quality of Service (QoS) in IoT networks.

Behnam Farzaneh and Ali Kamil Ahmed (2019) proposed the Multi-Criteria Routing Protocol (MC-RPL) to improve QoS in IoT networks. MC-RPL used the VIKOR method to provide Multi-Criteria Decision Making. VIKOR used multiple routing metrics of both link and the node including the lowest delay, ETX and remaining energy to choose the best parent. The network simulator known as Cooja network was utilized to evaluate the efficiency of MC-RPL in IoT network. Performance parameters such as End-to-End Delay (E2ED), Packet Delivery Ratio (PDR), Average Energy Consumption (AEC), and Throughput were used to evaluate MC-RPL's efficacy. The evaluation results depicted that MC-RPL improved QoS in an IoT network of 30 nodes maximum network size. In a small-scale IoT network, the traffic is distributed evenly thanks to multiple routing measures criteria. However, the effectiveness of this method must be assessed in a more extensive network, and further testbed tests must be carried out to yield more insightful results.

RPL fails to handle data transmission in high traffic load as the neighbor nodes become congested when the child nodes forward voluminous data simultaneously. Therefore, the authors proposed Burst and Congestion Aware metric for RPL (BCA-RPL) metric to prevent congestion in the neighboring nodes(Altwassi et al. 2019). The BCA-RPL algorithm determines a ranking by examining the children of a given node. Hop count is used as the key ranking parameter to ensure that the workload is distributed evenly across all nodes. Contiki OS was used to implement the method, and the Cooja network simulator was used to test it. BCA-RPL outperformed standard RPL by reducing Packet loss ratio and energy consumption. This algorithm may increase control plane messages when the child nodes switch parents. To avoid control overhead, the algorithm should consider bandwidth allocation during link creation.

Bhandari and Cho (2019) presented effective parent selection for the RPL routing algorithm in an advanced metering infrastructure (AMI) network. The routing scheme had the aim of constructing a balanced routing topology to avoid network congestion. The scheme used multiple routing metrics to compute the route that results in a balanced routing topology that balances the network, avoids congestion, and improves network reliability. The child count and residual buffer were responsibly adopted as node metrics and ETX for link metrics. The routing system was built in Contiki OS and tested against RPL and EC-OF in the Cooja network simulator. The routing

scheme improved the packet delivery ratio and throughput and reduced the packet loss ratio and latency in the AMI network. The validity of the routing algorithm should be tested in a testbed environment, for a longer time. Other performance metrics such as control overhead should also be included to recommend it for other IoT applications.

The Expected Transmission Count and Parent Count RPL (ETXPC-RPL) protocol was proposed by (Altwassi et al. 2019). The protocol was regarded as the most effective load-balancing method in IPv6 to manage the burst traffic load. The method relied on the child count routing metric to distribute traffic, and the quality of communication was evaluated by monitoring the percentage of packets that made it to the sink node and the amount of energy utilized. The performance of ETXPC-RPL was evaluated in Cooja compared with RPL. The protocol outperformed RPL in the Packet delivery ratio, decreased the power consumption, and increased the lifespan of the network. This algorithm needs to be evaluated on other performance metrics to make sure that it is useful for RPL load balancing. Comparing the performance of ETXPC-RPL with other RPL-based protocols may contribute to identifying the optimal load-balancing algorithm for RPL in large networks.

To balance the load in the RPL protocol for WSNs, the scholars (Sampayo et al. 2019) presented the Load Balancing Parent Selection (LoBaPS) method. The algorithm optimizes the parent selection by utilizing the Wake-up Radio (WuR) nodes to forward packets. The packets are transmitted as a Wake-up-Request (WREQ) containing a node rank to initiate communication over WuR. Lobes algorithm keeps the network robust to node shutdown and requires less control overhead. Wake up-request procedure may result in delaying the transmission of the packet.

Ghaleb et al. (2019) suggested a Load Balancing Aware Objective function (LBSR) to distribute work fairly across the RPL protocol. To determine the optimal parent, the objective function computes the number of available children in the parent. The parent selection strategy reduces the herding effect inside a network. LBSR outperformed RPL in terms of Packet delivery ratio, load distribution manner, and energy consumption of the nodes to forward the packets. This algorithm was based on the same network size of the nodes, but different packets sent per minute. To assure its efficacy, it must be examined across various network sizes.

The load balancing approaches proposed minimize the congestion problem in RPL and enhance network lifetime. The authors (Zaroor 2020) proposed an objective function Enhance-Minimum Rank with Hysteresis Objective Function (E-MHOF) to optimize network lifetime. Through the application of several routing measures, the objective function optimized path selection, parent selection, and child minimization. Expected Transmission Count (ETX), RSSI, and Nodes Residual Energy have been utilized to identify best parent for transmitting the packets. The maximum Remaining energy of the node and minimum ETX value of the path is used to choose the best path from the child node. The congestion on the parent nodes was alleviated by minimizing the child nodes through RSSI reference and the threshold value. In the Cooja network simulator, E-MHOF's performance was assessed against default objective function (MRHOF). The traffic load balancing was successfully implemented, and it reduced the congestion on the parent nodes and enhanced the network lifetime. The performance of E-MRHOFF was limited in a low network of 25 nodes in a large simulation area. This algorithm needs to be evaluated in a large IoT network to ensure its effectiveness in balancing the load and find out how it controls the network based on control overhead performance metrics.

The energy balancing objective function (EBOF) was introduced by Prakash and others (Rana et al 2020) to address the load balancing issue in the RPL protocol. The energy balancing was designed to minimize the congestion in the parent nodes and reduce packet loss. The algorithm operated with reliability routing metric ETX and child count to choose the reliable parent node. The child nodes selected the parent node with fewer child nodes using the path with fewer retransmission rates. The procedure maintained the power consumption on the parent nodes and congestion. EBOF outperformed LB-OF, ALABAMO, and MRHOF IoT networks. The algorithm was limited to a fixed network of 40 nodes, it prolonged the network lifetime by balancing energy. Many performance metrics can be used to verify the EBOF's effectiveness in different network scenarios.

Seyfollahi and Ghaffari (2020) proposed a lightweight load-balancing and route-minimizing solution for routing protocol for low-power and lossy networks. The algorithm maintains the node's energy, increases the network's lifespan, and equalizes traffic loads across all of the communication links in the system. The parent selection method was utilized to select a preferred

parent based on the objective function. It controlled the parent switches to avoid routing overhead. The results were analyzed in the Cooja network simulator based on latency, energy consumption, and packet loss. In light traffic, L2RMR balanced the traffic distribution effectively but we are not aware of the performance of the algorithm in heavy traffic. For RPL in heavy traffic, the parent selection and congestion avoidance contribute to a control overhead that depletes the resources such as power on the nodes.

Congestion in IoT network scenarios has become a serious issue that affected network reliability, the authors (Bouzebiba and Lehsaini 2020) came up with the idea of implementing a new objective function that upgrades the bandwidth for better packet transmission in Io Multimedia Things (IoMT). The objective function was named as Free Bandwidth Objective function for RPL (free BW-RPL). The algorithm balanced the load distribution by calculating the free bandwidth in a distributed order and switching the paths when it reaches the congested path to mitigate the path congestion. It was built in Contiki and tested against the standard RPL objective function in the Cooja network simulator. It provides the shortest path with free bandwidth to reduce delay in the network, and energy consumption and improve throughput together with the packet delivery ratio. The algorithm can be trusted for all IoT scenarios after testing in real testbed experiments with large network sizes generating different traffic.

The RPL nodes in the IoT network forward non-uniform traffic. The problem is that some of the parent nodes have moderate traffic while others have heavy traffic, which causes congestion. Musaddiq et al. (2020) mentioned the necessity of balancing the load distribution in heterogeneous traffic distribution. They decided to propose a Queue and workload based on RPL (QWL-RPL) routing metric. Combining routing metrics was proposed to address shortcomings in standard RPL objective functions. RPL protocol's efficiency in a heterogeneous network was enhanced by QWL routing metrics by choosing a trustworthy route to send packets to the sink node. QWL reduced network latency and traffic overhead while boosting packet reception rates. However, an essential performance indicator to take into account while assessing routing protocols is the energy used by the nodes.

Kim et al. (2020) have proposed a Power Control Routing protocol for low-power and lossy

networks as the mechanism for managing transmission power to reduce load imbalance difficulties and hidden terminal concerns in RPL-heavy networks. The method added a new parent selection process that took into account the relative RSSI value of the parent candidate node and the adaptive RSSI threshold. In order to balance the load and address the concealed terminal issue, PC-RPL adaptively modified RSSI thresholds. Without compromising dependability, it reduced hidden terminal and link congestion by minimizing the transmission power in accordance with the reference RSSI and transmission outcomes. The algorithm was implemented on a real multi-hop testbed and evaluated in a fixed network of 49 nodes. Its performance was evaluated and compared with RPL and QU-RPL. PC-RPL improved throughput during link congestion mitigation, and it outperformed QU-RPL in terms of PRR, latency, and improved energy efficiency under heavy traffic. PC-RPL was never tested on different network densities to find its efficiency in low network, medium and large networks.

Pereira et al. (2020) proposed the network interface average power consumption metric (NIAP) for the RPL protocol to balance the load distribution and minimize the energy consumption in LLN. The NIAP metric was responsible for energy cost in the nodes, enabling the nodes to transmit the packets at limited energy. This was achieved by enabling the child nodes to choose less congested parents and fewer energy paths to improve the network lifetime. The NIAP metric was evaluated in a simulated environment of homogeneous hardware nodes in the random topology. NIAP was evaluated in several network sizes. It performed better than the ETX metric and it performed very well in LLNs. It improved the network lifetime by 24%, the packet delivery ratio by 1%, and depreciated the average end-to-end delay by 13%. The NIAP metric implementation is simple, but not specific to a certain RPL objective function. The routing metric needs to be checked based on how it performs in control messages in an IoT network.

Energy-efficient load-balanced RPL for IoT(EL-RPL) network has been proposed by (Witwit and Idrees 2020) to improve network lifetime and avoid network congestion among the parent nodes. EL-RPL balanced the load by utilizing the parent metric count and observing the remaining energy on the parent nodes. The performance of EL-RPL was evaluated in the OMNET++ simulator in different network densities. EL-RPL improved network lifetime by mitigating the energy during routing, reducing the control overhead in IoT networks. The algorithm proved to be a reliable load-

balancing routing algorithm in IoT networks, however, its efficiency needs to be evaluated in real network scenarios with many devices.

Load balancing capacity(LBC) and network lifetime (NL) in RPL for IoT networks have been optimized (Kumar and Hariharan 2020). Dual Context-based Rooting and Load balancing in RPL for IoT networks (DCRL-RPL) was introduced to overcome routing overhead and packet loss caused by unbalanced load distribution in RPL protocol. In DCRL-RPL, load balancing was improved by the use of a number of procedures, such as grid construction to create various levels of grids that would balance the load distribution and maximize network longevity. Grid head nodes were chosen using the Random Walk Ranking (RWR) method, which is based on a ranking system. The selected head node grid must be a node that would be able to perform great in heavy conditions. The grid's master node uses a reputation-based scheduling system to allocate work to its subnodes. This strategy prevents congestion during the transmission of data within each grid. The defined dual context-oriented objective function was used to identify the best parents for both sensitive and non-sensitive data. The data types are classified by Adam deep neural network (ADNN). To transmit vulnerable data, an optimization-based objective function (O2F) was designed and adopted by the spotted hyena optimizer (SHO). MOORA objective function (MOF) was created to operate with the MOORA algorithm to identify the optimum parent node to route data packets for non-sensitive data transmission. The adaptive trickle algorithm was also proposed for the optimization of control messages. The effectiveness and efficiency of DCRL-RPL in IoT networks were measured by packet loss ratio, end-to-end delay, routing overhead, and load balancing capacity. It was evaluated on the network simulator 3.26 (NS3). The DCRL-RPL algorithm outperformed all the load-balancing routing algorithms based on RPL that were compared within the IoT network. The other RPL-based load-balancing routing algorithms use routing metrics to balance the load distribution while DCRL-RPL did not use link metrics which may cause weak links for data transmission. The algorithm must be evaluated in a testbed to get a better analysis.

The RPL protocol can also benefit from partitioning the network into clusters in order to more evenly distribute the load. To prolong the life of IoT networks, Sennan et al. (2020) presented a cluster tree-based routing protocol called CT-RPL. In order to build the cluster, select the most

trustworthy node to serve as the cluster leader, and design the path that connects all of the cluster's child nodes to the leader. The Euclidean distance technique was used to group the nodes into a cluster. Each cluster had a cluster head that was responsible for communicating with the gateway. Therefore, the cluster head should have enough energy to perform the data transmission effectively. The game theory approach was utilized to choose the cluster head. The route between the cluster heads and cluster members was constructed based on the Residual energy ratio (RER), and queue utilization (QU) and ETX routing metrics. The evaluation's findings showed that CT-RPL is dependable for the network since it increases packet loss, lowers power consumption, and extends network lifetime. The CT-RPL effectively balanced the load in the IoT network, however, the algorithm may raise the routing overhead in the cluster members when the network becomes complex.

Stability-Aware Load balancing for RPL (SL-RPL) is proposed by (Babulak 2020) to balance load distribution and mitigate frequent parent changes that occur during congestion avoidance routing algorithms based on RPL. SL-RPL balanced the load by enabling child nodes joining the topology to select less congested parent nodes. It was evaluated in Contiki OS in comparison with ALABAMO, TAOF, and RPL algorithms. SL-RPL outperformed ALABAMO, TAOF, and RPL. It mitigated packet loss and power consumption to enhance the network performance in IoT networks. The delay of the packets was not considered an important parameter in network performance. Evaluating the performance of SL-RPL in high traffic would be beneficial to the ongoing problem of imbalanced load in RPL protocol.

Sebastian (2020) suggested using a Child Count-based Load Balancing in RPL (Ch-LBRPL) to detect load imbalance in control messages of the RPL protocol. It is presented as a means of improving upon current load-balancing methods. Although RPL shows great promise in low-traffic networks, it falls short in high-traffic networks since many nodes forward many DIO messages and it is therefore unable to optimize parent selection. The primary goal of the Ch-LBRPL algorithm was to decrease the average transmission rate of control messages. The algorithm balances the load in three phases namely DODAG, DAG, and Multi DAG Structure. To ensure that each parent node had the same number of children, DAG employed a child count mechanism. In DODAG where the nodes are restricted to forwarding packets to the parents. In

multi DODAG the algorithm balances the load in multiple sink nodes, the algorithm balances the load in large networks where the RPL has two or more sink nodes. The load is balanced in the root node or gateway in the third phase. Multi DODAG load balancing is used in the root node. The information about the new node (RPLInstanceId, DODAGID, node address, and, node Id) is saved each time it sends DIO to the preferred parents. The Ch-LBRPL algorithm was built in the Contiki operating system and analyzed using the Cooja network simulator. The algorithm improved the stability of the DAGs and minimized the parent switching of the child nodes, minimizing the energy for processing by reducing control messages. However, the mechanism minimizes the packet delivery ratio also, and the time to arrange the child nodes. The child count settings are insufficient to balance the load distribution among the nodes, as the daily growth in communication devices has led to an increase in network traffic.

The researchers and developers are researching an effective load-balancing algorithm for RPL in IoT networks while mitigating energy consumption. The Load Balancing and Energy Efficient RPL (LoB-RPL) routing algorithm have been proposed by (Arunachalam and Nallamothu 2021) to promote efficient data transmission with better data aggregation. The algorithm improved MRHOF and trickle algorithm to balance load distribution in large network scenarios of IoT networks. The effectiveness of the LoB-RPL over RPL protocol was measured in terms of packet delivery ratio, packet loss, throughput, and control overhead using the Cooja network simulator. Throughput, packet delivery ratio, overhead, and packet loss were all improved upon by LoB-RPL compared to RPL.

A load balancing time-based (LBTB)-RPL routing algorithm distributed the traffic effectively within the nodes by considering the nodes with high remaining energy as parent nodes (Yassien et al 2021). The transmission of packets through the high remaining energy of neighbor nodes and parent nodes assisted in improving the network performance. LBTB's performance has been evaluated in the Cooja network simulator compared to LBSR and default RPL in IoT networks. LBTB improved the network lifetime by reducing the delay of packets during simulation, as well as reducing power consumption. The algorithm balanced the load but needs improvement so that it will also improve the ratio of delivered to the destination. The low packet delivery ratio may be caused by control overhead which was also ignored in the network. There is a possibility of high control messages while the nodes are looking for a reliable neighbor node during data transmission.

The congestion avoidance routing algorithm was proposed by Shreyas and other authors in Shreyas et al. (2021). The algorithm optimizes the path of selection by introducing fuzzy logic. The algorithm named Congestion Avoidance uses Fuzzy Logic to discover the best path in the 6LoWPAN network (CAFOR). The parent selection problem was modeled using the Fuzzy weighted sum method. The combination of routing metrics was introduced to pick the efficient route for transmitting data. However, the algorithm can select non-congested paths to improve network lifetime. CAFOR uses a combination of Buffer occupancy, RtMetric, and ETX routing metrics to compute the rank of a node. It was implemented in Contiki and simulated in Contiki to evaluate its performance compared to OHCA and QURPL. CAFOR mitigated network delay and improved network performance with less energy consumed during data transmission. The algorithms need to be tested in a testbed to ensure their efficiency and evaluate the control overhead in the network. The table 3.2.1 summarizes the single path load balancing routing algorithms proposed for RPL in different network scenarios with different aims.

*Table 3.2 1: Single path RPL-Based Load balancing routing algorithms*

<b>Proposed Algorithm</b>	<b>Aim of the Algorithm</b>	<b>Utilized Routing metrics</b>	<b>Tools</b>	<b>Performance metrics</b>	<b>Drawbacks</b>
QU-RPL(Kim 2015)	Investigate the load balancing and congestion problem in the RPL routing algorithm in LLNs	Hop count ETX	Tiny OS Testbed	End-to-end delay Packet delivery ratio Routing overhead	Node congestion may arise and cause a buffer overflow and affect the network performance.
Congestion -Aware RPL for 6LoWPAN (Al-Kashoash et al. 2016)	Balance the load to avoid congestion on the parent nodes in the 6LoWPAN network	Buffer occupancy ETX	Cooja	Packet loss Throughput Packet delivery ratio Energy consumption	Fixed number of Nodes, ignored the remaining energy metric during load balancing
Improved Trickle for RPL(Goyal and Chand 2018)	Load balancing to reduce power consumption in Low power networks	Trickle timer parameters modified.	Contiki Os Cooja network simulator	Packet delivery ratio Average power consumption Average energy consumption	No impact on the packet delivery ratio. Does not provide reliable links for data transmission

<b>Proposed Algorithm</b>	<b>Aim of the Algorithm</b>	<b>Utilized Routing metrics</b>	<b>Tools</b>	<b>Performance metrics</b>	<b>Drawbacks</b>
LB-OF (Qasem 2018)	Balance the load to mitigate network overhead	Child node count	Contiki OS 2.7, Cooja network simulator	Control overhead	Partial load balancing is provided by amending topology formation
EL-RPL (Sankar and Srinivasan 2018)	Maximize the network lifetime	Battery Depletion Index (BDI) ETX	Cooja network simulator	Packet delivery ratio Network lifetime End-to-end delay	Limited to low-cost network. Power consumption may increase during the parent changes.
CLRPL (Taghizadeh et al. 2018)	Mitigate packet loss to improve RPL performance in Heavy and dynamic load scenarios.	CARF ETX Remaining Energy (RE)	Cooja network simulator	Packet loss ratio Energy consumption Network overhead	Evaluated in a heavy network scenario, in other network congestion notification increases control overhead
LB-RPL(Parsaei et al. 2018)	Introduce load balancing to alleviate congestion in RPL	Buffer utilization	Cooja network simulator	Packet delivery ratio End-to-end delay Throughput	Poor links utilized to transmit packets.
lbRPL(Sebastian and Sivagurunathan 2018)	To quantify load balancing in RPL for LLNs	ETX Parent count Remaining energy	Contiki OS Cooja	Network stability Network lifetime Packet delivery ratio Control overhead	Limited to a low-cost network. Control messages may increase in a different network scenario.
EN-RPL(Singh and Chen 2019)	Improves the network stability through the parent selection mechanism	Composite Efficient Routing ETX	Contiki OS Cooja	Queue-loss Packet loss End-to-end delay Energy consumption	Control overhead in the large network will increase,

<b>Proposed Algorithm</b>	<b>Aim of the Algorithm</b>	<b>Utilized Routing metrics</b>	<b>Tools</b>	<b>Performance metrics</b>	<b>Drawbacks</b>
BCA-RPL (Altwassi, Pervez and Dahal 2019)	Improve the network lifetime in burst traffic IoT networks	Parent count Hop count.	Cooja	Packet loss Power consumption Packet delivery ratio	Increase control messages when parent nodes indicate congestion.
Proposed(Bhandari and Cho 2019)	Construct a balanced routing topology	Buffer occupancy Child count ETX	Contiki OS Cooja network simulator	Packet delivery ratio Packet loss ratio Latency	Simulated for a short time, low network. Ignored stability metric power consumption
ETXPC-RPL (Altwassi et al. 2019)	Measure communication quality and optimize network quality in a burst traffic load	ETX Parent count	Cooja network simulator	Packet delivery ratio Power consumption	Limited to a fixed network density. The algorithm was evaluated on a few performance metrics
LoBaPS (Sampayo et al 2019)	Balance the parent's nodes data to Overcome the parent selection issue in RPL in WSN	Hop count Parent selection	Cooja network simulator	Network lifetime Packet delivery ratio	High power consumption on the nodes. Unstable network resulting from using poor links
LBSR (Ghaleb et al 2019)	Optimize parent count in RPL	Hop count, ETX	Contiki OS	Packet delivery ratio	Herding effect problems.
FLEA-RPL(Kumar and Hariharan 2020)	Maximize the network lifetime in LLNs	Residual Energy ETX	Cooja network simulator	Parent changes Average delay Remaining energy Packet delivery ratio	Unstable network topology determined by high parent changes
E-MRHOF (Zaroor 2020)	Avert congestion in parent nodes to improve network lifetime in IoT network	ETX RE RSSI	Contiki OS Cooja network simulator	Power consumption Latency	Limited to a low network traffic Few performance metrics utilized to verify the performance of E-MRHOF

<b>Proposed Algorithm</b>	<b>Aim of the Algorithm</b>	<b>Utilized Routing metrics</b>	<b>Tools</b>	<b>Performance metrics</b>	<b>Drawbacks</b>
EBOF(Rana et al 2020)	To enhance the RPL performance in LLN environment	Child count ETX	Contiki OS Cooja	-Average energy consumption Network lifetime Average parent change	High frequent parent changes may be dangerous to power consumption when network density increases.
L2RMR (Seyfollahi and Ghaffari 2020)	Avoid thundering herd problems in LLNs	No specific routing metrics	Contiki OS Cooja network simulator	-End-to-end delay -Packet loss ratio Energy consumption	In large network scenarios. High packet loss will affect the network due to the retransmission of data
FreeBW-RPL (Bouzebiba and Lehsaini 2020)	Optimize path selection	ETX Energy Free bandwidth	Contiki OS Cooja network simulator	Packet delivery ratio Throughput End-to-end delay Energy consumption	Effective in low network traffic, in large traffic thundering herd effect, will affect the network.
QWLRPL (Musaddiq et al 2020)	Achieve reliable paths in heterogeneous traffic to optimize network performance in IoT networks.	Queue-utilization	Contiki OS Cooja network simulator	Packet reception ratio Packet loss ratio Average delay Total overheads	Limited to a simulation environment and no testbed evaluation
PC-RPL(Kim et al 2020)	Mitigate hidden terminal and load imbalance issues	RSSI	Tiny OS Testbed	Packet delivery ratio End-to-end delay	Buffer overflow is the root cause of packet loss.
NIAP (Pereira et al. 2020)	Optimize energy on the RPL nodes in WSNs	Hop count	Contiki OS Cooja	Network lifetime Energy consumption Packet delivery ratio	High parent changes will result in high control overhead and power consumption

<b>Proposed Algorithm</b>	<b>Aim of the Algorithm</b>	<b>Utilized Routing metrics</b>	<b>Tools</b>	<b>Performance metrics</b>	<b>Drawbacks</b>
DCRL-RPL (Kumar and Hariharan 2020)	To alleviate RPL energy problems in IoT networks.	ADDN	NS3	End-to-end delay Network lifetime Routing overhead Load balancing capacity Packet drop ratio Packet loss ratio	Results in a network buffer overflow
CT-RPL (Sankar et al. 2020)	Improve the quality of service. Provide maximum network lifetime in IoT network	RER QU ETTX	Contiki OS Cooja network simulator	Energy consumption Packet loss ratio Parent changes Delay	Frequent parent changes in a real network cause damage to power consumption. Buffer overflow chances are high in cluster heads
SL-RPL (Babulak 2020)	Alleviate parent oscillation to preserve network stability in the IoT network	ETX PTR Hop count	Contiki OS Cooja	Energy consumption Parent changes Packet loss	Reliable in low network sizes, parent nodes may suffer from high consumption of energy when network size increases
Ch-LBRPL(Sebastian 2020)	Alleviate parent switching to improve network lifetime in IoT network	Child count	Contiki OS Cooja	Energy consumption Control overhead	Partial load balancing in low network traffic. Limited to a simulator
LoBRPL (Arunachalam and Nallamotheu 2021)	Mitigate network congestion in parent nodes to stabilize the network in the IoT network	Hop count ETX Workload	Contiki OS Cooja	Packet delivery ratio Packet loss Throughput Control overhead	Low network density Low simulation time. Further verifications must be conducted in different network scenarios and testbed

<b>Proposed Algorithm</b>	<b>Aim of the Algorithm</b>	<b>Utilized Routing metrics</b>	<b>Tools</b>	<b>Performance metrics</b>	<b>Drawbacks</b>
LBTB(Yassien et al 2021)	Controls time intervals to build a balanced RPL network topology for IoT network.	Neighbor count RE	Contiki OS Cooja	Power consumption Convergence time Packet delivery ratio End-to-end delay	Excessive packet loss due to network overhead that occurs during DODAG construction
CAFOR (Shreyas et al 2021)	Finding the optimal routing path in the 6LoWPAN network	BO Hop count ETX RtMetric	Contiki OS Cooja	Throughput Power consumption Goodput End-to-end delay	Control overhead may rise in a large network traffic
C-Balance (Royae et al 2021)	Conserve battery to improve network lifetime in fixed and mobile nodes	ETX RE Hop count Children count	Contiki OS Cooja	Packet delivery ratio End-to-end delay Power consumption Network lifetime	Packet oscillation when cluster node's battery depleted
AMRRPL (Royae et al 2021)	Save power and improve network lifetime in IoT network	LQI BO RE SNR	Contiki OS Cooja	Packet delivery ratio Network lifetime Convergence energy	Network latency was neglected in the study. Limited to a simulation environment
FTC-OF (Hassani et al 2021)	Minimize the parent changes to stabilize the network	RSSI Hop count FTM	Contiki OS Cooja	Power consumption Parent changes	Limited to a simulation study, low network density. Network delay may affect the heterogeneous network
CBR-RPL (Shirbeigi et al. 2021)	Provide reliable data routing in IoT network	NQ PQ	Contiki OS Cooja	Energy consumption Packet delivery ratio	Unstable paths among the nodes raise packet loss

The scholars (Fatemifar and Javidan 2021) presented the optimal load balancing and energy

efficient load balancing algorithm, a new load balancing clustering approach for load balancing in RPL protocol. The C-based algorithm divides the network into clusters to apply balanced load distribution and improve the network lifetime. The algorithm applies residual energy, hop count, and ETX routing metrics and counts the number of children per parent to implement load balancing. Cooja, a network simulator, was used to model the implementation of C-balanced in Contiki. C-balanced algorithm performance was compared to that of MRHOF, OF0, and QU-RPL in both mobile and non-mobile nodes. Each cluster had an independently deployed set of nodes, with the cluster head being responsible for aggregating data from the subordinate nodes and transmitting it to the gateway. Algorithms' effectiveness was evaluated with respect to three metrics: power consumption, packet delivery rate, and latency in the network. In both scenarios, C-balance outperformed QU-RPL, MRHOF, and OF0, but resulted in high overhead and network delay. Clustering a network is a great load-balancing approach, however, it needs to consider buffer occupancy routing metric to avoid control overhead caused by the retransmission of data packets.

Non-uniform data distribution in the network results in network congestion of the parent nodes in a large IoT network scenario. To overcome this situation, The authors made adjustments to the original version of RPL to ensure that the IoT network's load was distributed evenly (Royae et al. 2021). Automata-Ant colony based multiple recursive RPL is the name given to the updated version of RPL that distributes work across nodes in a network to alleviate bottlenecks. The algorithm used three different steps to balance load distribution. It presented the use of learning automata to manage queues and control network congestion. Queue management has the purpose of avoiding congestion by preventing buffer overflow. The algorithm provided an efficient parent selection mechanism to avoid the bottleneck in the parent nodes.

Multiple routing metrics play a vital role in balancing traffic distribution, Forwarding Traffic Consciousness objective function has been proposed for better transmission of packets in RPL in an IoT network. FTC-OF (Hassani et al. 2021) is a novel objective function that combined three routing metrics to choose the optimal parent node. The combination of Radio Signal Strength Indicator (RSSI), Hop Count, and Forwarding traffic metric balances the load distribution by finding the effective parent node, and shortest path and reducing energy consumption by

minimizing frequent parent changes. The effectiveness of FTC-OFs was assessed using the Cooja network simulator; it balanced the load effectively in the IoT network.

Poor network performance got the attention of researchers like Shirbeigi et al (2021) who presented an RPL routing algorithm that takes clusters and packet drops into account to extend the life of IoT networks and reduce data loss. The algorithm categorizes the network into clusters in the same DODAG root, where the sink node receives data from cluster heads. The cluster heads are special to have more buffer space to accommodate a large amount of data. When the nodes are unable to find the node with free space to accommodate more packets, it can join another cluster in the network. CBR-RPL improves the packet delivery ratio more than QURPL and RPL, mitigating power consumption. However, CBR-RPL is efficient in balancing load in heterogeneous networks although the control overhead increases when the nodes advertise themselves as the available nodes and during frequent changes of the nodes moving from one cluster to another cluster. The movement of the node may result in unnecessary network delay.

The single path RPL load balancing routing algorithms effectively balanced the load in different network scenarios with various traffic patterns. The algorithms performed load balancing using more than one routing metric such as integrating measures of individual links and nodes. Some of the load-balancing routing algorithms result in control overhead when the child nodes are searching for reliable parent nodes (Mamdouh et al. 2016; Santiagor et al. 2018; Chen et al. 2020; Witwit and Idrees 2020). Following is a discussion of the multiple path RPL load-balancing routing algorithms proposed by several of the authors for use in different types of networks.

### 3.3 Multipath RPL-based Load balancing routing algorithms

Effective load distribution in RPL routing protocol has been an issue when RPL is routing in large network traffics. RPL effectively transmits data in low-traffic network scenarios. However, in IoT network scenarios it performs poorly because the nodes transmit data utilizing either link quality or hop count. The parent nodes are easily congested, lose packets and deplete energy in large traffic scenarios, which results in network instability and unreliability. As a means of enhancing RPL's functionality across a variety of massive-scale network use cases, several researchers designed

routing algorithms that utilize multiple routes, and sink nodes to transmit data in different network scenarios (Farooq et al. 2017; Lodhi et al. 2017; Zhu et al. 2017).

The algorithm CA-RPL (Congestion Avoidance Multipath Routing algorithm based on RPL) was proposed by the authors (Tang et al. 2016). The protocol focused on reducing the network delay. It proposed a routing metric DELAY\_ROOT for RPL, the metric reduces the average rate of delay toward the routing node. The suggested protocol reduces network congestion by permitting the dispersal of heavy traffic to various pathways. The protocol used Delay root and EXT routing metrics to direct the traffic to the parent node and provide efficient traffic distribution. The CA-RPL protocol was evaluated against RPL using the EXT routing metric in a Z1 mote network environment, using the Contiki OS and the Cooja network simulator. The simulation findings indicate that the CA-RPL protocol reduces packet loss and delay by 20% and 30%, respectively, compared to the original RPL. The protocol came up with a strategic way of avoiding the congestion, and not focusing on the actual algorithm of balancing the load to avoid network congestion.

The authors created a multi-parent RPL protocol energy-balancing routing algorithm to extend the network lifetime and stabilize the network (Iova et al. 2017). The algorithm utilized multiple parent nodes to ensure that all the packets reach their destination on time. Reliable parent nodes were selected based on their remaining energy. A DAG that calculated the network lifetime was built using the Expected Transmission Count (ETX) and ELT (Expected LifeTime) metrics. The algorithms promoted choosing the preferred parent, and reduced network delays. Algorithm performance was measured using the WSN simulator in terms of both successfully delivered packets and overall power usage. The algorithm perfectly balanced the energy in Wireless Sensor Networks. In large traffic scenarios, Energy balanced-RPL may pose issues of frequent parent changes when the parent nodes are congested.

Researchers have proposed a new protocol called Multipath Extension of RPL (M-RPL) to alleviate congestion during routing and extend the life of networks. M-RPL distributed the load over the network through multiple nodes to balance the load distribution in the RPL protocol. The technique utilized node parameters such as Packet delivery ratio and buffer occupancy to balance

the load distribution among the nodes, while DIO messages were used to notify where the congestions occur. Latency, throughput, and power consumption were used to assess M-RPL's performance. The efficiency of the algorithm was measured by comparing its operation on a grid topology to that on a random topology. In grid topology, M-RPL balanced the load effectively. The evaluation results depict the improvement of throughput, decrease in power consumption, and packet loss. Therefore, M-RPL successfully balanced load distribution over default RPL. Although multiple-path routing raises frequent parent changes that result in high traffic overhead and packet loss, the algorithm used only node metrics to detect congestion that may raise an issue of using poor links. Whenever the packets are transmitted via a poor link, packet loss might be a problem and degrade network performance.

The Multipath routing mechanism for RPL (MRPL) was proposed to maximize the network lifetime by providing multiple paths between the nodes (Moghadam et al. 2014). This algorithm minimizes the cost of data transmission by allowing the nodes to have multiple paths to a sink node. This method minimizes both power usage and packet loss. MRPL balanced the load within the nodes using a heuristic load distribution mechanism (HeLD) that assists in equalizing energy between the competing nodes. The program also extended the communication range to equalize power usage among nodes operating at the same level. The MRPL algorithm was implemented in OMNET++ and simulated in MATLAB to evaluate its performance in the network. In numerous simulations, MRPL outperformed standard RPL with a 38% improvement in the network lifetime, 19% in packet delivery ratio and 51% in throughput. MRPL effectively balanced the load distribution in a small network which was displayed in a 50m network area. The algorithm utilized different approaches to minimize the cost of routing in LLN networks, however, the algorithm never reveals how the network was controlled during the data transmission. More investigations can be carried out for MRPL such as evaluating the performance in a heterogeneous IoT network and finding out about the delay of the network in a large simulation area.

The authors proposed an energy-aware multipath routing protocol on the Internet of Things that balances the load distribution based on composite metrics. The protocol was named Energy and Load aware RPL (ELaM-IoT) (Bagherifard et al. 2020). The algorithm balanced the load distribution with multiple routing metrics that include battery depletion, remaining energy, hop

count, and Link Expiration Time (LET) in the network.. ELaM-IoT performance was compared with ADRM-IoT and ERGID approaches based on network lifetime and remaining energy performance metrics. The performance took place in the NS-3 network simulator within the network of 15 packets/sec. The results show that ELaM-IoT improved network lifetime with high remaining energy. The algorithm must maintain the network delay which seems to be the problem affecting the network lifetime.

RPL routing protocol suffers from imbalanced load most often in dense network scenarios where the child nodes broadcast the packets via the available route, lowering the packet reception ratio and slowing the network. To address the imbalance of network traffic in the RPL protocol, adaptive multipath traffic load has been suggested (Zhu et al. 2017). AMTL-RPL balanced the energy between the reliable parent nodes, which assisted in improving the network lifetime. AMTL-RPL allocated the packets in the right channels to avoid network delay, it used the catch method to ensure that all packets reach the destination. Effective load balancing was achieved by the method using ETX routing metrics and remaining energy. In AMTL-RPL, the nodes transmit packets through multiple routes. This may raise issues of duplicating the data, high control overhead, and frequent parent changes when the preferred parent node's energy is depleted.

In the heavy network of IoT, RPL nodes transmit heterogeneous data traffic which burdens the sink (border router) node. An overloaded border router performs badly because it misses packet transmission deadlines. These authors have proposed a new version of the RPL protocol that includes network prefix and load balancing (RPL-NPT-LB) in an effort to fix the problem of failed border routers (Foubert and Montavont 2019). The RPL-NPT-LB divide the traffic into two parts, allowing the network to use multiple border router to balance the load distribution effectively. The border routers cooperate when making a global decision about data transmission between the child nodes to avoid transmission errors. The IPv6 reconfiguration address occurs when the node changes the border router mitigated by IPv6-to-IPv6 network prefix translation. The RPL-NPT-LB routing algorithm was built in Contiki OS and assessed on an actual testbed. The approach increased network lifetime and decreased end-to-end delay while improving energy consumption. The algorithm's efficiency was assessed over a low network. However, the effectiveness of the RPL-NPT-LB routing algorithm must be measured on large-scale networks and using other

performance measures.

The Multi-class Multipath routing protocol for low power and lossy network with energy balanced optimal rate has been proposed by Farahbakhsh (2021). The algorithm is known as M2RPL, and it distributes the load across the nodes using an energy-balanced optimal rate algorithm (EBORA). Balancing energy while choosing a reliable parent node has been the trusted method to optimize the quality of service (QoS). Simulation findings demonstrated that M2RPL was able to evenly distribute the load in a wireless sensor network with 60 nodes. The approach extended the lifespan of the network and reduced latency between nodes, but it also increased computational complexity, produced redundant packets, and caused packet collisions.

Parent selection optimization in RPL is conducted with an objective function utilizing its ETX and Hop count routing metrics. Child nodes select the parent nodes based on the shortest path or strong link quality; There are occasions when the majority of nodes choose the same node to be their parent when they realize that the congestion changes the parent node and moves to the same parent node, which results in herding effect. Therefore, authors Sebastian and Sivagurunathan (2018) proposed a multi DODAG procedure to optimize parent selection in the RPL protocol. As a result of combining numerous DODAGs into a single RPLInstance, network resilience was increased, and the network's capacity to distribute its load evenly across devices was enhanced. Multi DODAG routing procedure resulted in low control overhead, low power consumption, also enhanced packet delivery ratio in IoT smart networks. Nonetheless, the multiple DODAGs routing procedure can increase network delay, The network delay can occur when the sink node on another DODAG fails, the parent nodes will shift the focus and join the alternative DODAG, there will be a little delay and more burden on the parent nodes of the resisted DODAG sink node.

In large network scenarios, some of the nodes are unable to reach the sink node due to imbalanced load sharing in the RPL protocol. The Multi-sink Load Balancing Objective Function (MSLBOF) has been suggested by (Abdullah et al. 2021) to extend the design of MRHOF of RPL. MSLBOF was intentionally proposed to boost the RPL's performance in a large network. Distributing the load evenly was achieved by the use of a composite routing metric. MSLBOF utilizes memory metrics to monitor DAAG status for selecting the best sink node and includes Sink Load Balancing

Option (SLBO) in a single DIO control message. Memory consumption and DAG size are used to determine a node's rank in MSLBOF. However, the performance of MSLBOF on a large scale determines that it is 85% reliable to the network, and mitigated packet drop. MSLBOF has been evaluated only on 100 nodes in a simulation, then it needs to be evaluated in a testbed and investigate the power consumption. This algorithm may raise an issue of high-power consumption on the sink nodes; they have a lot of packets to process. During routing, routing loops may occur since the proper parent selection procedure was ignored.

Recently the authors proposed an algorithm that balances the load distribution and caters to buffer utilization in the RPL protocol. The Weighted Random Forward RPL for High Traffic and Energy Demanding Scenarios (WRF-RPL)(Acevedo et al. 2021) has been proposed to balance load distribution in WSN scenarios. It performed the proper load balancing by utilizing multiple routing metrics for different network traffic sizes and network sizes. The algorithm was evaluated in Contiki OS over ETXPC-RPL and lbRPL load-balancing routing algorithms. Fortunately, WRF-RPL improved network lifetime and PDR in different traffic scenarios reducing control overhead and power consumption. However, the time consumed for data transmission is not considered although it is an important metric to estimate the network cycle.

The Multi sink load balancing mechanism has been proposed for WSNs to overcome the single point of failure in the RPL protocol (Onwuegbuzie et al. 2022). MSLBM was designed to utilize more than one sink node for data transmission to optimize energy consumption and balance the load distribution effectively. The algorithm adopted the improved least connection method to optimize load balancing across the sensor nodes. The effectiveness of MSLBM was assessed across a range of network loads, from 20 to 200 nodes. The simulation results identified MSLBM as an optimal load balancing for RPL in WSN because it prolonged the network lifetime. The evaluations were based on average latency, power consumption, and packet delivery ratio. However, MSLBM neglected to adjust the formation of the DODAG structure which may result in high control overhead and oscillation of packets among the sink nodes.

Balancing load distribution considering multiple paths, and multiple sinks have been the best idea to speed up the process with the drawbacks of increasing frequent parent changes when The

dependable parent or sink is overloaded. The primary goal of most Multipath RPL load balancing routing algorithms is to increase the reliability and lifespan of the network. These algorithms allow the neighbor nodes to transmit data using multiple paths, or multiple sink nodes to improve the lifespan of the network. The most utilized routing metric is energy balancing which assists in improving the lifespan of the network, however, the ETX routing metric is considered the stability metric on RPL protocol. Most of the multipath RPL load balancing ignored the link metric which may cause the problem in a long run. Poor link quality results in high packet loss and poor network quality. Table 3.3.1 depicts the summary of proposed multipath RPL-based load-balancing routing algorithms.

*Table 3.3 1: The summary of multiple path load balancing algorithms for RPL*

<b>Proposal</b>	<b>Aim</b>	<b>Utilized routing metrics</b>	<b>Evaluation tools</b>	<b>Evaluation metrics</b>	<b>Drawbacks</b>
CA-RPL (Tang et al. 2016)	Improve monitoring events in emergency scenarios. Reliable data transmission in WSN	Delay -root Time factor	Cooja	-Packet loss ratio -Latency -Throughput	Neglected power consumption, Congestion avoidance metrics will raise floods of control messages in a large network
ELT (Iova et al. 2017)	Stabilize the network also improve the lifespan of the network	ETX Expected Life Time (ELT)	WSNet simulator	PDR, network stability through preferred parent changes, CCDF, and energy consumption	Limited to a low fixed network size High packet drops during parent oscillations

<b>Proposal</b>	<b>Aim</b>	<b>Utilized routing metrics</b>	<b>Evaluation tools</b>	<b>Evaluation metrics</b>	<b>Drawbacks</b>
M-RPL (Lodhi et al. 2017)	Mitigate network congestion and enhance throughput	Congestion detection and buffer size	Cooja Network Simulator. Random and Grid Topology	End-to-end delay, Throughput and average energy consumption	Reliable in low network traffic scenarios and simulation results
Minimum cost load-balanced multipath RPL (Moghadam et al. 2014)	Improve network lifetime and reduce transmission cost	Heuristic load distribution algorithm, transmission range	OMNET++ and MATLAB	Network lifetime, throughput, and packet delivery ratio	Utilize poor links that result in excessive energy consumption during frequent parent changes and packet drops
ElaM-IoT (Bagherifard et al. 2020)	Improve network performance by balancing the energy in the IoT network.	The remaining energy, link Expiration time (LET), hop count, and load batter deletion index (DBI)	NS3	Mean Remaining energy and network lifetime	Neglect resourceful performance metrics that identify the reliability of the algorithm.
AMTL-RPL (Zhu et al. 2017)	Stabilize the network, improve network lifetime	Energy balance	Contiki os Cooja network simulator	Residual energy, end-to-end delay, network lifespan, and packet delivery ratio.	High packet loss due to poor links utilized for data transmission.
RPL-NPT-LB (Foubert and Montavont 2019)	Balance the load effectively to improve energy consumption and bandwidth	Bandwidth	Contiki OS Testbed	End-to-end delay per node, Energy consumption	Performance was limited to low network traffic. Neglected the link quality which may

<b>Proposal</b>	<b>Aim</b>	<b>Utilized routing metrics</b>	<b>Evaluation tools</b>	<b>Evaluation metrics</b>	<b>Drawbacks</b>
	utilization				result in packet loss in large network traffic
M2RPL (Farahbakhsh 2021)	Find a way to enhance service quality and equalize data transmission.	Energy balance optimization-based traffic distribution	NS3	Network lifetime, Packet delivery ratio	More control messages generated result in control overhead
Multi DODAG in RPL (Sebastian and Sivagurunathan 2018)	Improve RPL performance in IoT for Smart city applications	The remaining energy of the node	Cooja network simulator	control overhead, Power consumption, network lifespan and packet delivery ratio	Requires extra memory and data communication which results in high power consumption in the nodes.
MSLBOF (Abdullah et al 2021)	Address the load balancing problem across the multiple sink nodes to stabilize the network	ETX, Memory utilization, and sink selection	Cooja Network simulator	Packet loss ratio, Packet delivery ratio and Control overhead	High power consumption when nodes change the sink nodes
WRF-RPL (Acevedo et al. 2021)	Avoid network congestion	Remaining energy and parent count	Cooja network simulator	Control overhead, Network lifespan, packet delivery ratio and power consumption	Delays the network when the packets are waiting on the queues to the reliable parent node.
MSLBM (Onwuegbuzie et al. 2022)	To extend the robustness of RPL to prolong the operational lifetime	Power and Memory of the sink	Contiki os Cooja simulator	Packet delivery ratio Latency Power consumption	Neglected node and link metrics may result in high control overhead and packet loss

Multipath, multi-sink, and multiple DODAGs load balancing routing algorithms have some drawbacks that affect the network negatively. The multiple-path selection procedure causes the duplication of packets, the privileged nodes may transmit the same packets effectively while other nodes are still waiting. When the sink node depleted the power and dies, packet retransmission occurs while the neighbor nodes are looking for the alternative sink node, and more resources are wasted during that process. Retransmission of packets leads to network delay, packet loss, and overuse of power in the nodes, therefore in this study, we neglect the multipath or multi-sink load balancing routing algorithms because IoT devices transmit a million of data in a short time. Multipath RPL-based load-balancing routing algorithms solve node unreachability problems, improve network lifetime and mitigate congestion in parent nodes. However, it causes high communication which consumes more power, generates more control messages, and drops the packets due to poor link qualities.

### 3.4 Summary

The RPL protocol is a standard routing protocol for low-power and lossy networks, because of its ability to effectively transmit data in the network. Congestion and power loss are issues for RPL due to the complexity of the network, which has a negative impact on the performance of the network. Because the load balancing method was not taken into account during the construction of RPL, there is poor network performance. Several scholars are concerned about finding a reliable load-balancing routing algorithm for RPL protocol. Scholars proposed the load balancing that is effective for their studies such as low power and lossy networks, wireless sensor network, and others (Farooq et al. 2015; Kim 2015; Tall et al. 2015; Bhandari et al. 2018; Ben Aissa et al. 2019; Jothikumar et al. 2019). The RPL-based load balancing routing algorithm utilizes different procedures, multiple paths, multiple sinks, and single sink nodes to transmit data. All these algorithms balance the load partially and they raise some drawbacks. Multipath load-balancing routing algorithms have a risk of duplicating the data, and multiple sink load-balancing routing algorithms have a risk of data oscillation, high power consumption, and control overhead when the reliable parent node is congested. IoT applications transmit sensitive data that need to be transmitted effectively. Reliable RPL load balancing is mandatorily required to transmit data effectively in IoT networks. The existing RPL-based load-balancing routing algorithms are effective in certain conditions such as LLNs, WSN, 6LoWPAN, Heavy and burst IoT and others.

Their performance was based on many performance measures, and each of them was assessed in various network densities. Finding the most reliable RPL load-balancing routing algorithm for IoT networks is challenging due to the disparities in performance of the existing RPL load-balancing routing algorithms. This study is intended to find effective RPL-based load balancing for IoT network scenarios by evaluating the performance of the existing RPL load-balancing routing algorithms in the same IoT network. From the exploration of the existing RPL-based load-balancing routing algorithms, we decided to select ETXPC-RPL, lbRPL, and CLRPL load-balancing algorithms and evaluate their performance in IoT networks of various network densities. These RPL load-balancing routing algorithms previously outperformed RPL in different network scenarios based on different performance metrics. In this study, the performance of RPL-based load balancing routing algorithms is evaluated based on power consumption, Packet delivery ratio, network delay, RT-metric, and control overhead in the Cooja network simulator. After evaluating the performance, we proposed enhancements to overcome the gaps that were discovered during the simulation in the IoT network.

The implementation of the load-balancing algorithms is detailed in the following chapter. The chapter described the objective functions and routing metrics utilized to balance the traffic distribution in ETXPC-RPL, lbRPL, and CLRPL and described how the routing metrics incorporated the enhancements of the selected routing algorithms. It also described the operating system used for implementing the routing algorithms. The performance metrics used for evaluation and network simulator are detailed in the next section.

# Chapter 4

## Research Methodology

Many industries, in recent times, have adopted the use of automation where things communicate without human intervention. As more and more services move online, there is a corresponding rise in the number of connected devices. The heterogeneous traffic generated by devices connected to the network contributes to the network's high traffic volume. The high data transfer rate produces network congestion, which hinders data transmission between devices. Therefore, the routing protocol is highly required to govern data transmission among IoT devices.

The Internet Engineering Task Force designed a routing protocol for Low power and lossy network (RPL) to improve data transmission in low-power devices. However, this protocol is unable to manage data transmission effectively in large traffic networks because it does not balance the load distribution effectively. Numerous authors have suggested various load-balancing routing algorithms based on the RPL protocol, these algorithms functioning differently in various network conditions. Since the performance of the algorithm depends on the network environment, it is difficult to find an effective RPL-based load-balancing routing algorithm for IoT network scenarios. This study aims to find an effective and efficient RPL-based load-balancing routing algorithm for IoT networks.

To conduct this study and achieve the objectives, the design science research method (DSRM) has been adopted which consists of different steps for solving a problem. (Pefferers et al. 2007; Venable et al, 2017) This chapter describes in detail the methods used to conduct this study.

Design science research methodology (DSRM) has a rigorous procedure for designing artifacts such as models, constructs, procedures, and instantiations. They may also contain social innovations or novel characteristics of social, technological, or informal resources. The DSRM method gives a conceptual model of research output characteristics. The schema is tabulated below. DSRM works effectively to achieve the goal of the study by applying the guidelines in table 4.1.

Table 4.1: Design Science Research Methodology (Peppers et al 2007)

Research Steps	Concern	Output to the next step	Entry Point
1. Identify problems and motivate	Define Problem Display value	Inference	Problem centered initiation
2. Define the objectives of the solution	What would be achieved by the superior artifact?	Theory	Objective centered initiation
3. Design and development	Artefact	acquiring knowledge	Design and Development centered initiation
4. Demonstration	Find a suitable context. Use the artifact to address the problem	Metrics Analysis Competence	Client/ Content initiation
5. Evaluation	Observe the effectiveness and efficiency of the algorithm in the network. Iterate the design	Disciplinary expertise	
6. Communication	Academic publications		

## 1. Choosing research method

In choosing a research method suitable for your study, it is essential to describe whether your study is driven by solving a certain problem or achieving certain objectives to address the problem. In this research, we have many objectives to fulfill to find an effective RPL-based load-balancing

routing algorithm for IoT networks. Therefore, we aligned this study with an objective-centered approach to DSRM.

### Objectives centered approach

Poor load balancing on RPL routing protocol results in poor data transmission and unstable networks in IoT. The need for effective RPL-based load-balancing routing algorithms for IoT networks leads to the implementation of RPL-based load-balancing routing algorithms in IoT networks. To find the effective RPL-based load-balancing routing algorithm for IoT networks, the study must propose the possible RPL-based load-balancing routing algorithms, compare their performance in IoT networks, perform analysis and identify the effective RPL-based load-balancing routing algorithm for IoT networks. The diagram in fig.1 depicts the different steps undergone to conduct this study under DSRM.

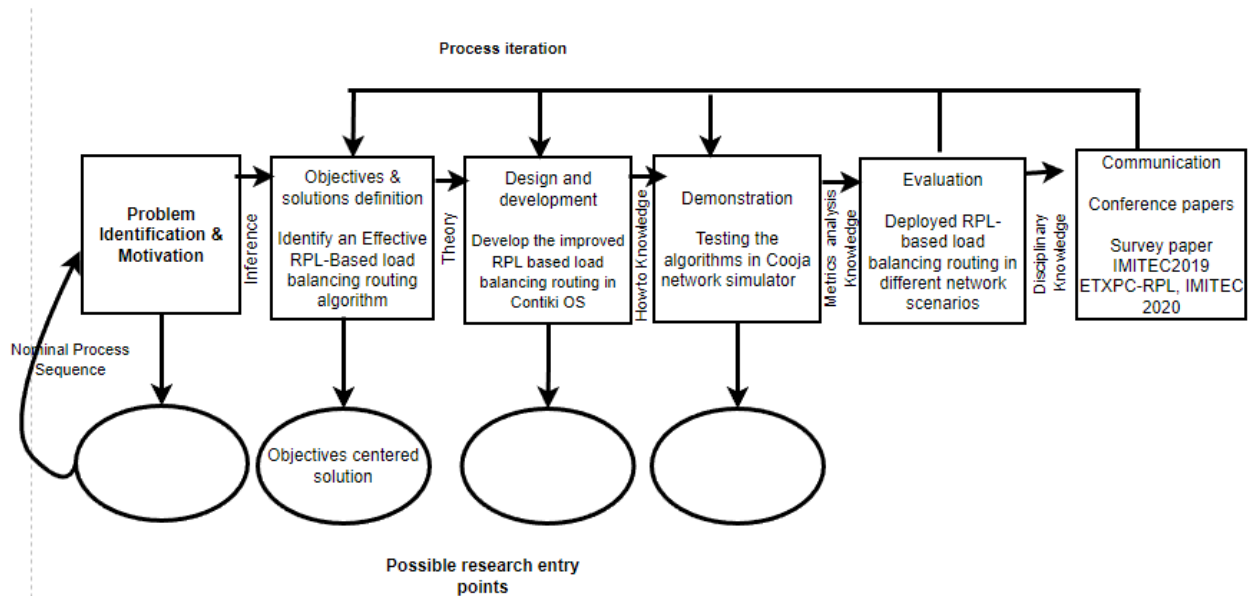


Figure 4.1: DSRM process Model which is an objectives-centered approach

## 4.1 Problem identification and Motivation

The RPL routing protocol manages data transmission with a single routing metric which does not balance the load distribution during routing. RPL protocol was standardized for low-power and lossy networks. Due to imbalanced load distribution, the IoT network faces obstacles such as high network latency, significant packet loss, and high energy use. Existing RPL-based load-balancing routing algorithms lack standardization of routing metrics and performance metrics. The performance of the existing algorithm is examined in various network sizes, thus confusing the

identity of the effective RPL load-balancing routing algorithm for IoT networks. The Internet of Things devices are responsible for transmitting critical information such as medical results, student records, and financial records, therefore effective data transmission mandatory. To accomplish the purpose of this study, we have a few objectives set to address the questions raised by the research.

## 4.2 The objective of the solution

The objectives of this study were to understand the state of the art of the load balancing routing algorithm in IoT networks. The literature review was conducted by reading published articles based on RPL load-balancing routing algorithms. The literature provided insight into the performance of the existing RPL-based load-balancing routing algorithms and identified the gap in the research. After identifying the gap, the second objective was to design an effective RPL-based load-balancing routing algorithm for the IoT network which was addressed by implementing the proposed RPL-based load-balancing routing algorithms in the IoT network.

The last objective was to identify or recommend an effective RPL-based load-balancing routing algorithm for IoT networks. This objective leads to the simulation of the implemented routing algorithms and the evaluation of their performance in different network sizes.

## 4.3 Design and development

To find the effective RPL-based load-balancing routing algorithms, this study designed three RPL-based load balancing algorithms for IoT networks named Enhanced Context aware and load balancing for RPL(ECLRPL), Buffer Occupancy load balancing routing algorithm (BLRPL) and Enhanced ETXPCRPL load balancing routing algorithm. The algorithms were developed on the Contiki OS platform which has complete RPL implementation. The sections below cover the implementation platform of the proposed RPL-based load balancing routing algorithms in contiki os.

### 4.3.1 Contiki Operating System

Contiki OS is an open-source operating system that provides a limited, lightweight, and transportable environment for micro sensor networks and embedded Internet of Things devices. The development of OS is still ongoing with the contribution of developers from different companies such as CISCO, Things Square, and Atmel (Thingsquare 2014). The software that runs on it is written in the industry-standard C programming language. Contiki was created to link

devices with limited resources to the internet. Contiki offers a network stack with standard IP protocols such as TCP/IP, UDP, and HTTP to add on the new low-power standards like IPv4, IPv6, 6LoWPAN, and CoAP to support IoT functionality (Padmaja et al. 2017). The operating system is used in both commercial and non-commercial applications to offer services such as smart city devices and industrial monitoring devices. There are two parts to the Contiki system: the core and the loaded program. The core programs include the Contiki kernel, a program loader, a language run-time, and a communication stack with communication hardware drivers.

The Program loader retrieves the necessary code for running the operating system and other applications from the network stack or an external storage device like EEPROM and then loads it into (random-access memory) RAM. Two kilobytes of random-access memory (RAM) and forty kilobytes of read-only memory (ROM) make up the Contiki setup. Since it is built on an event-driven kernel, Contiki also enables multi-threading, and supports protothreads, which are a lightweight stack-less thread-like architecture.

Contiki offers a collection of practical libraries for memory management and linked list operations. The operating system provides layer-specific components. It contains various files, including `up-nds6.c`, that provide neighbor detection modules under the "Contiki/core/net" directory. It also includes the ability to generate and analyze all ND ICMPV6 messages. The `up-ds6c` supports the creation and processing of ND conceptual data structures as well as stateless address autoconfiguration. The OS offers the routing modules, which are spread across multiple files in the "Contiki/core/net/rpl" directory. They are properly divided up into distinct categories according to the services they offer. For example, `rpl-icmp6.c` provides functionality for packing ICMPV6 messages, while `rpl-dag.c` contains the functionality of forming Directed Acyclic Graphs (DAGs). Contiki allows the modification of files or the creation of new protocols related to your study. In this study, we modified RPL to balance the load distribution. IoT enables the embedded devices to communicate from the local mesh network to the internet, however, Contiki provides reliable communication of IoT devices with the real world.

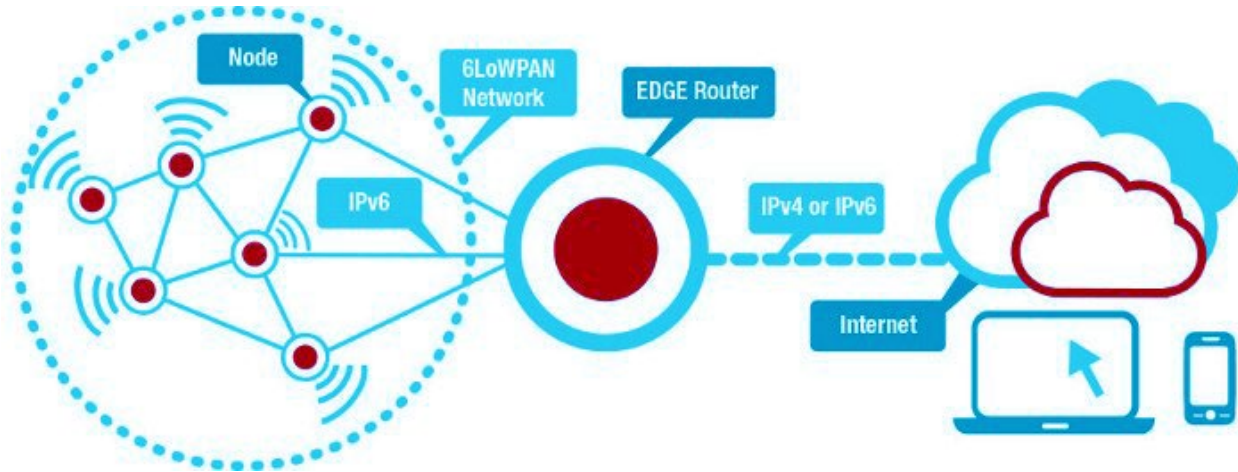


Figure 4.3.1.1 Communication components in Contiki Operating System (Sebastian and Sivagurunathan 2018).

Contiki consist of APIs used to develop enhanced RPL-based load-balancing routing algorithms for IoT network.

The source code in Contiki 2.7/core/net/rpl was modified to implement each RPL-based load balancing routing algorithm. The table 4.3.1 describes the functionality of each file in Contiki OS.

Table 4.3. 3: RPL files in the Contiki operating system

File	Functions
Makefile.rpl	Include modules to be utilized in RPL operation
rpl-dag.c	Supports the functions that initialize and maintain the DAG
rpl-of0.c	Using the hop count routing metric to implement the objective function zero
rpl-mrhof.c	Implementation of minimum rank with hysteresis objective function with ETX routing metric
rpl-icmpv6	Responsible for ICMPV6 control messages implementation
rpl-etx-header.c	The extension headers for RPL implementation in Contiki are stored here
rpl.c	After DAG is created, the routes are maintained in this file
rpl-timers.c	Contains the useful implementation of the trickle timer algorithm and parameters for updating the network

- **Enhanced CLRPL(ECLRPL)**

Enhancing a Context-aware load-balancing routing algorithm is proposed by extending the existing Context-Aware and Load balancing routing algorithm for RPL (CLRPL) which was proposed by Taghizadeh (Taghizadeh et al. 2018). The algorithm proposed objective function and routing metrics for RPL to route packets in a balanced manner in heavy dynamic traffic networks. CLRPL proposed a Congestion Avoidance Objective Function (CAOF) that performed better routing with congestion avoidance routing metric CARF by avoiding congestion on the parent nodes and root node. CA-OF performs the nodes rank computation based on the context of the node: it prevents the thundering herd phenomenon by gradually shifting from the high-rank value to further down the path. The routing algorithm proposed the routing metric CARF to perform load balancing. The CARF routing metric provides a better parent selection mechanism. The child nodes select the preferred parent based on the node's residual energy, and link quality (ETX value) and examine the queue channel. The routing algorithm performs load balancing while avoiding the routing loops to improve network reliability in IoT networks.

ECLRPL algorithm performs routing using the congestion-aware objective function CA-OF. This objective function operates with three reliable routing metrics to ensure that the load distribution is balanced in the network. The routing metrics include remaining energy on the parent nodes. This enables the child node to transmit the packet to the parent node with high remaining energy to avoid sending packets to the nodes that are out of power. The second metric is congestion avoidance which indicates when the parent node is congested, and the last metric is buffer occupancy. The buffer occupancy alerts the child nodes about the remaining buffer capacity of the parent before the child nodes select a parent node as a preferred parent. Combining the routing metrics is an effective way of balancing the load distribution in an IoT network. The routing metric's function is described below:

- I. Congestion Avoidance Objective Function (CAOF)

The objective function analyses the context of the next node to compute the rank of the current node and it chooses the actual rank by shifting gradually from the rank of high value; in that manner, it avoids the thundering herd phenomenon. The APIs utilized to build CAOF are found in table 4.3.2 which describes the use of each API.

Table 4.3. 4: RPL APIs used to create the new objective function CAOF.

RPL API	Function
reset(dag)	This is utilized to reset the state of the objective for a certain DAG: the function is powerful for global repairing of the DAG
neighbor_link_callback (parent, status, etx):	This is utilized for receiving link layer neighbor information.
best_parent(parent1, parent2)	This is used to find the best parent by comparing two-parent nodes based on the OF and returning the best one
best_dag(dag1,dag2)	This finds the best DAG among two DAGs based on OF and returns the best DAG
calculate_rank(parent, base_rank)	This function is responsible for calculating the rank value of the node based on the rank of the parent and base rank
update__metric_container(dag)	In a specific DAG, it is responsible for updating the metric container for outgoing DIOs.

The Context-Aware and load balancing routing algorithm were proposed to balance the load distribution in IoT of fast-speed sensors. CLRPL proposed a new objective function Context-Aware Objective function (CAOF) which performed load balancing utilizing a routing metric Context-Aware Routing Metric. CARF chose the optimal parent depending on the available power and the number of nodes in the queue channel. The fig.4.3.1.2 depicts the files in RPL library with the objective function CAOF.

In contiki 2.7/core/net/rpl

```

user@instant-contiki: ~/contiki-2.7/core/net/rpl
File Edit View Search Terminal Help
user@instant-contiki:~/contiki-2.7/core/net/rpl$ ls
Makefile.rpl  rpl-conf.h    rpl-ext-header.c  rpl-mrhof.c    rpl-timers.c
rpl.c         rpl-conf.h~  rpl.h             rpl-of0.c
rpl-caof.c   rpl-dag.c    rpl-icmp6.c      rpl-private.h
user@instant-contiki:~/contiki-2.7/core/net/rpl$

```

Figure 4.3.1.2: RPL files in Contiki 2.7 with the rpl-caof objective function

CONTIKI\_SOURCEFILES += rpl.c rpl-dag.c rpl-icmpv6.c rpl-timers.c \ rpl-caof.c rpl-etx-header.c

## II. Context-Aware Routing Metric (CARF)

The routing metric considers the remaining power and queue utilization of the parent chain towards the root recursively while lessening the effect of upstream parents as it gets further down the path.

The queue channel is responsible for congestion mitigation. This metric examines the space of the neighboring nodes based on their buffers and is referred to as the buffer occupancy ratio of node I to indicate the number of packets in the I's queue out of the total size. The node's congestion is easily indicated by the buffer occupancy mechanism and enables the nodes to forward packets toward the idle or loaded parent node. The channel utilization is represented by the equation (4.3.1):

$$CU_{(i)} = N_P Q_{(i)} / QSz_{(i)} \quad (4.3.1)$$

Where the  $CU_{(i)}$  is an element  $[0,1]$ .  $N_P Q_{(i)}$  is the number of packets in the queue of node I and  $/QSz_{(i)}$  is the total queue size of i.

## III. Residual Energy (RE)

The Residual Energy (RE) proposes that the new node examine the level of energy of its desired parent. It's a routing metric that's used to figure out which node to send data to based on how much power it has left. RE is defined as the difference between maximum energy  $MxE(i)$  and the energy consumption  $Ec(i)$  by node i. However, the energy consumption can be approximated based on the power consumed by the node in each of its possible operating states: CPU (when the radio is off, but the system is still active and processing data), LPM (low power mode or sleep mode), reception (Rx) and transmission (Tx). There is a module for energy estimation in Contiki which is known as Energise. This module is utilized to determine the length of time the system has been in various states.

Cooja emulator offers the Unit Disk Graph Medium (UDGM)- distance loss an elementary radio model which models the transmission range among the nodes in the network. The network nodes are represented graphically by disks, and their potential for interconnection with one another is represented by the intersection of these disks, which indicates the feasibility of point-to-point

transmissions between the nodes. In contrast, the chance of receiving packets diminishes as the distance between two nodes grows. The remaining energy of the node was calculated using the equation in (4.3.2)

$$(RE_i) = (M_x E_{(i)} - Ec_{(i)}) \quad (4.3.2)$$

#### IV. Parent Selection

The approach manipulates CARF and other efficient routing parameters (Buffer occupancy) to prevent packet routing loops. The goal of combining routing metrics in IoT networks is to evenly distribute network traffic. This enables the RPL protocol to be suitable for voluminous data flooded by IoT devices from different IoT applications. However, this algorithm ignored the status of the node which may be an issue with buffer overflow. Enhancement includes a new routing measure that checks the buffer capacity of the parent nodes prior to selecting the parent node as the preferred node:

The algorithm operates with four routing metrics in CAOF objective function.

Preferred parent

If a parent has high remaining buffer+ high residual energy+ strong link quality and high bandwidth

Choose as preferred parent.

Else

Choose another node that meets the requirements.

All these routing metrics were utilized to balance the load distribution by transmitting the data packets with a reliable link to a parent node with a less occupied buffer capacity.

- **Buffer occupancy Load balancing for RPL(BLRPL)**

Sebastian and Sivagurunathan (2018) proposed an efficient RPL load-balancing routing metric (lbRPL). This routing metric balances the traffic distribution in Low Power and Lossy Networks (LLNs). For efficient data transmission, the lbRPL distributes load efficiently among the routes. The algorithm uses Load Balancing Indicator (LBI), remaining energy and ETX routing metrics

to select the optimal path to an effective parent node. It provided efficient packet distribution and improved network lifetime in LLN. However, in this study, the algorithm was amended with the Buffer occupancy routing metric to improve the buffer size of each node for the IoT network.

The nodes choose to balance the load in the root by considering the best parent node as a neighbor node. The best parent is the node with the highest remaining Energy ( $E_r$ ), the highest link quality (ETX value), and the highest Buffer Occupancy (BO) parameters. The remaining energy dictates how long a network node will remain connected. Higher remaining energy defines the node's ability to remain on the network longer, whereas lower remaining energy dictates the node's shorter lifetime. However, the nodes with high remaining energy can be reliable parent nodes for transmitting data to the root node by the current energy on the node from the consumed energy. The remaining energy can be determined by using the equation (4.3.3):

$$ER = \frac{ER_{remaining}}{E_{initial}} \quad (4.3.3)$$

BLRPL chooses as the optimal parent node the node with the highest amount of available energy. The flow chart in fig 4.3.1.3 explains the parent selection criteria of BLRPL in IoT networks. The goal of combining routing metrics is to achieve more even load distribution and provide more efficient routes for data delivery. The packets are sent from the children to the DODAG root via the chosen parent. Each node selects the parent with high remaining energy to make sure that the node has enough energy, checks the available capacity of the node to carry the packets to avoid network congestion, the node also performs a link quality check to ensure that data packets are not lost due to a weak connection.

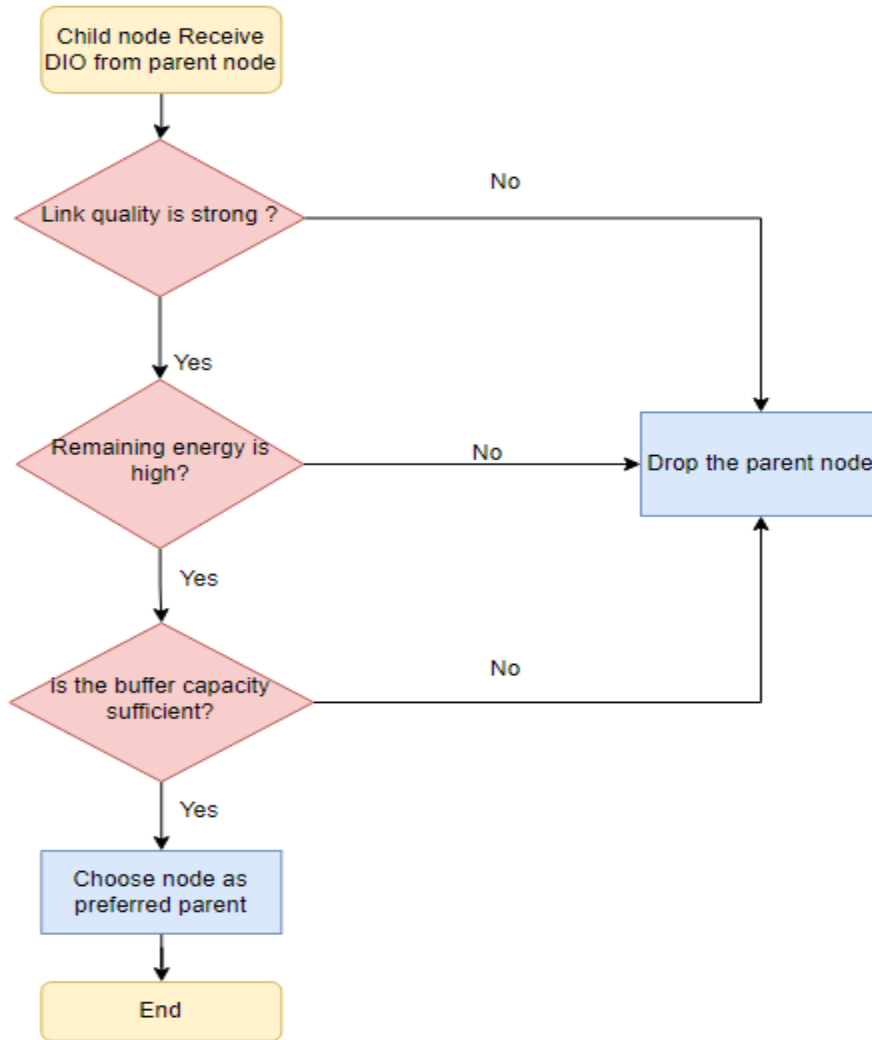


Figure 4.3.1.4: BLRPL flow diagram

- **Enhanced ETXPC-RPL**

The Internet of Things consists of an enormous number of devices that transmit heterogeneous data simultaneously. The data transmitted by IoT devices exponentially increases every time, so it is difficult to manage its movement from one device to another device. Designing a reliable routing procedure can be a solution to data transmission in an IoT network. Enhanced RPL load balancing is an extension of the ETXPC objective function for RPL which was proposed to balance load distribution in burst scenarios (Altwassi et al. 2019). ETXPC modified the MRHOF objective function of RPL to balance the load in burst traffic scenarios. Therefore, Enhanced ETXPC proposes a node metric buffer occupancy to improve data transmission in IoT networks. EN-

ETXPC-RPL relies on parent nodes with high buffer space to accommodate more packets, especially in the high network. The child nodes select the best parent with enough buffer size and the best link quality (ETX). The BO routing metric provides the upgrade in the node's capacity and indicates the space available to accommodate incoming packets. If the parent node has enough packets, it rejects the request coming from the child node to avoid network congestion. The algorithm also uses the ETX routing metric and counts the physical number of nodes currently joined by the parent. To guarantee effective packet transmission, the path to the parent node is selected based on the best available link. The ETX routing metric plays a crucial role in maintaining network reliability.

#### Parent count mechanism

The parent of the child node was selected based on ETX, Buffer occupancy, and Parent count. Expected Transmission Count (ETX). It is the number of times a packet must be sent out before it is finally received at its intended location. The ETX value provides insight into the health of the network and can be used to make an estimation of its reliability. The network's link stability is indicated by a low ETX value, that results in fewer retransmissions of packets and the nodes consume less power and decrease the delay. We utilized equation (4.3.4) to calculate the value of ETX:

$$ETX = \frac{1}{Df * Dr} \quad (4.3.4)$$

Df determines a forward delivery ratio, which quantifies the likelihood that a neighbor will really get a packet. While Dr is an acknowledgment probability that guarantees the packet was successfully delivered to its destination (Fatemifar and Javidan, 2021b).

Buffer occupancy is the node metric responsible for the node status to indicate the congestion on the node. It is determined by the total buffer size minus the packets available in the buffer which is calculated as using equation(4.3.5):

$$BO = Btotal - occupied\ buffer \quad (4.3.5)$$

When the nodes transmit packets to the congested parent nodes, the new coming packets cause a buffer overflow, and they end up lost. Therefore, buffer overflow is here to mitigate packet loss by allowing leaf nodes to select less congested parent nodes. The parent count approach also allows child nodes to see the available nodes in the intended parent node. If there are more than three

child nodes that would like to send packets to the desired parent, the child node immediately switches parents. The modifications have been done to balance load distribution in RPL for IoT networks because RPL does not have effective load-balancing parameters. EN-ETXPC-RPL used triple routing metrics to overcome load balancing in IoT networks because many deployments can be made which have different expectations. Load balancing is conquered with Buffer Occupancy, Parent count, and ETX routing metrics.

However, in this section, we utilized Contiki 2.7 OS to implement the selected load-balancing routing algorithms in the IoT network scenario. We changed the objective function in Makefile.rpl to rpl-mrhof.c, to utilize the link quality metric ETX. In the Makefile the changes made look like the displayed fig. 4.3.1.4:

```

user@instant-contiki: ~/contiki-2.7/core/net/rpl
File Edit View Search Terminal Help
user@instant-contiki:~$ cd contiki-2.7/core/net/rpl
user@instant-contiki:~/contiki-2.7/core/net/rpl$ ls
Makefile.rpl  rpl-conf.h~    rpl.h          rpl-of0.c
rpl.c         rpl-dag.c     rpl-icmp6.c   rpl-private.h
rpl-conf.h   rpl-ext-header.c rpl-mrhof.c  rpl-timers.c
user@instant-contiki:~/contiki-2.7/core/net/rpl$

```

Figure 4.3.1.5: RPL files in Contiki 2.7.

```

CONTIKI_SOURCEFILES += rpl.c rpl-dag.c rpl-icmpv6.c rpl-timers \rpl-mrhof.c rpl-etx-
header.c.

```

We made changes in rpl-conf.h to change the objective function and add energy-based routing metric.

```

#ifdef RPL_CONF_OF

#define RPL_OF RPL_CONF_OF

#else

/*ETX is the default objective function. */

#define RPL_OF rpl_mrhof

```

```
#endif /*RPL_CONF_OF*/
```

To utilize the energy metric

In `contiki2.7/platform/sky/` defines the energy metric in `contiki-conf.h`

```
#define RPL_CONF_DAG_MC_ENERGY
```

In `rpl-conf.h`

```
#ifdef RPL_CONF_DAG_MC
```

```
#define RPL_MC_RPL_DAG_MC
```

```
#else
```

```
#define RPL_DAG_MC_ENERGY
```

```
#endif /* RPL_CONF_DAG_MC*/
```

After implementing the algorithms in Contiki OS, the algorithms were simulated in the Cooja network simulator to evaluate their performance in a simulated environment. The following section describes the demonstration of RPL-based load-balancing routing algorithms in IoT-simulated networks.

## 4.4 Demonstration

The algorithms developed in the Contiki operating system were simulated in the Cooja network simulator (Thomson 2016) which is an ideal tool for simulating wireless sensor networks. Simulation demonstrates how the nodes communicate in the network. The network was deployed from ten nodes up to one hundred nodes. Each network was simulated for 20 minutes and repeated ten times to find the average results per network size. The simulation was conducted to find the effective RPL-based load-balancing routing algorithm for IoT networks. The performance of the algorithms was evaluated based on the results obtained after every simulation using different performance metrics.

Implementation of the routing algorithms was the initial step that leads to the simulation of the algorithms in a network simulator that will assist in finding the effectiveness of routing algorithms in IoT. However, the implementation of RPL-based load balancing and its enhancements was conducted in Contiki OS that is a specialized piece of software made specifically for sensor

networks and other types of networked embedded devices. It is a highly regarded open-source OS for facilitating communication amongst Internet of Things gadgets and it can connect resource-constrained devices to the internet at a low cost (Padmaja et al. 2017). Contiki is a firmware for most popular sensor nodes like Zoletia-Z1, WisMote, and TSky Mote and supports IETF protocols for low-power IPv6 stack. Routing algorithms were developed in C language. To balance the load distribution on RPL, we modified the open-source codes of objective functions to satisfy the aim of the study. In our modifications we used, `rpl-config.h`, `rpl-obj`, `rpl-mrhof.c`, `makefile.c`, `udp-sink.c` and `udp-sender.c`.

The experiments were conducted in the Cooja network simulator provided by the Contiki operating system. Contiki is a simulator that is implemented in the java programming language. Contiki graphical user interface Cooja (Bagula and Erasmus 2015; Thomson 2016), Cooja performs more like an emulator because it can emulate real-life motes in the most realistic way possible. It has a high degree of accuracy. Cooja allows sensor nodes and the sensor nodes software to be written in C language.

Cooja is a wireless network and IoT applications network simulator that is bundled with Contiki OS. Cooja is rated as the best simulator to simulate any wireless sensors with its property. It has the flexibility to change the node's properties to implement researchers' ideas. It is a java-based simulator with a java Native Interface that develops application software in C language. Cooja provides various simulation mediums: each has unique characteristics, plugins, and tools. The unit disk graph medium is the most popular medium which uses signal strength to limit transmission ranges. Cooja has the advantage of being able to simulate the application software simultaneously at the operating system level, application (networking) level, and the instruction level for machine code. The simulation environment parts can be altered without changing a single main code of Cooja. The simulator has the flexibility of enabling the addition of pure java code nodes to simplify and speed up software development in certain scenarios. The pure java nodes are important for the efficient simulation of heterogeneous networks because they require less processing power and memory (Österlind et al. 2006). Plugins like the timeline, simulation visualizer, and Radio logger are used to run the simulations and observe the results. The simulated data is kept in an XML file with the ".csc" extension (Cooja Simulation Configuration). This file stores simulation-related data such as node locations, plugin, radio medium, random seed, , etc.

The simulator executes the Contiki programs, which are stored in a different directory and may also include a "project-conf.h" file that allows for the modification of RPL settings centrally. Cooja is the simulator that comes with the Contiki OS which was mainly developed for IoT devices communication. Therefore, Contiki allows the implementation of IoT routing protocols and Cooja assesses the effectiveness of the established routing protocols.

Three various levels of simulations can operate simultaneously in Cooja -Operating system level, Network level, and Machine code instruction level. For this study, several tools used during simulations include instant Contiki which is an Ubuntu image of Contiki OS with all tools installed.

The VMware player: The instant Contiki image was loaded into the VMWare player.

Wireshark is a protocol analyzer that is used as a packet sniffer to find the control messages generated in the network.

Jupyter notebook was used for statistical analysis to generate plots.

The operating system has tools and plugins that are useful for saving different network features used for the analysis of the results such as a simulation visualizer that provides the opportunity to visualize network traffic, radio environment, grid, and mote types as is shown in Fig. 4.4.1.

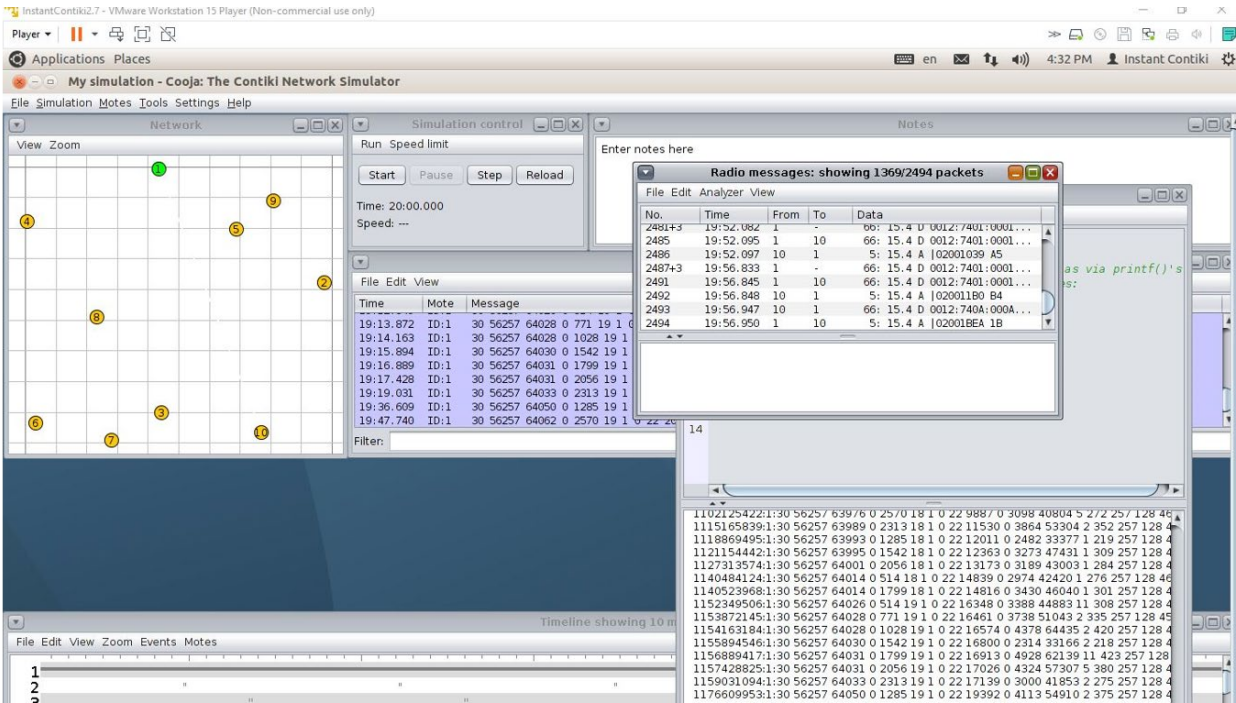


Figure 4.4.1: Cooja network simulation of 10 nodes randomly distributed in the network area.

The timeline provides a timeline of data transmission, turnaround time and acknowledgment time and it shows when the collisions occurred. There is a radio logger which captures the network packets in the pcap file and mote output which is beneficial for checking data flow and other messages. Fig 4.4.1 shows a snapshot of the Cooja tool being run in Contiki 2.7. The simulation is stored as an XML file with the extension 'csc' (Cooja simulator Configuration). This file contains important details regarding the simulation environment, the nodes and their placements, radio medium and random seeds.

Cooja has sophisticated tools for data collection from the nodes during simulation. Information from all of the nodes in the network that can potentially interact with the sink node is compiled in a single place: the sensor data collecting view. The results from the collect view were collected after the simulation has stopped. Many details are within the window which is used for simulation purposes such as the hops traveled by each node during the simulation, ETX values, power consumed by the nodes, duty cycle, etc. Fig 4.4.2 shows the collect view plugin with the results collected during the simulation of 50 nodes for 20 minutes.

Nodes	Node Control	Sensor Map	Network Graph	Sensors	Network	Power	Node Info	Serial Console											
<All>	Node	Received	Dups	Lost	Hops	Rtmetric	ETX	Churn	Beacon Interval	Reboots	CPU Power	LPM Power	Listen Power	Transmit Power	Power	On-time	Listen Duty Cycle	Transmit Duty Cycle	Avg I
1.1	20.20	16	0	0	1.000	256.000	8.000	0	18 min, 33 sec	0	0.567	0.146	0.593	0.052	1.359	3 min...	0.989	0.098	
2.2	21.21	13	0	0	2.000	384.000	16.000	0	16 min, 07 sec	0	0.472	0.149	0.418	0.029	1.068	2 min...	0.696	0.054	
3.3	22.22	16	0	0	1.000	256.000	8.000	0	18 min, 33 sec	0	0.546	0.147	0.461	0.033	1.187	3 min...	0.769	0.062	
4.4	23.23	13	0	0	2.000	384.000	16.000	0	16 min, 47 sec	0	0.504	0.148	0.452	0.037	1.140	2 min...	0.753	0.069	
5.5	24.24	11	0	0	3.000	512.000	24.000	0	14 min, 17 sec	0	0.354	0.153	0.403	0.018	0.927	2 min...	0.671	0.033	
6.6	25.25	16	0	0	1.000	256.000	8.000	0	19 min, 39 sec	0	0.528	0.148	0.485	0.020	1.180	3 min...	0.809	0.038	
7.7	26.26	14	0	0	2.000	384.000	16.000	0	16 min, 13 sec	0	0.559	0.147	0.439	0.027	1.173	3 min...	0.732	0.051	
8.8	27.27	16	0	0	1.000	256.000	8.000	0	18 min, 33 sec	0	0.516	0.148	0.445	0.012	1.120	3 min...	0.741	0.022	
9.9	28.28	15	0	0	2.000	384.000	16.000	0	15 min, 43 sec	0	0.529	0.147	0.468	0.050	1.195	3 min...	0.781	0.094	
10.10	29.29	14	0	0	2.000	384.000	16.000	0	14 min, 20 sec	0	0.481	0.149	0.455	0.094	1.179	3 min...	0.759	0.176	
11.11	30.30	17	0	0	1.000	256.000	8.000	0	19 min, 00 sec	0	0.406	0.151	0.413	0.016	0.986	3 min...	0.688	0.029	
12.12	31.31	17	0	0	1.000	256.000	8.000	0	17 min, 59 sec	0	0.623	0.145	0.778	0.170	1.715	3 min...	1.297	0.320	
13.13	32.32	11	0	0	3.000	512.000	24.000	0	14 min, 17 sec	0	0.373	0.152	0.431	0.029	0.985	2 min...	0.718	0.055	
14.14	33.33	17	0	0	1.000	256.000	8.000	0	17 min, 59 sec	0	0.515	0.148	0.450	0.018	1.131	3 min...	0.750	0.034	
15.15	34.34	15	0	0	2.000	384.000	16.000	0	15 min, 43 sec	0	0.515	0.148	0.459	0.038	1.160	2 min...	0.766	0.071	
16.16	35.35	14	0	0	2.000	384.000	16.000	0	14 min, 20 sec	0	0.499	0.148	0.434	0.028	1.109	3 min...	0.723	0.053	
17.17	36.36	16	0	0	1.000	256.000	8.000	0	18 min, 33 sec	0	0.519	0.148	0.442	0.015	1.124	3 min...	0.737	0.028	
18.18	37.37	14	0	0	2.000	384.000	16.000	0	15 min, 35 sec	0	0.485	0.149	0.431	0.029	1.094	2 min...	0.718	0.054	
19.19	38.38	13	0	0	2.000	384.000	16.000	0	16 min, 07 sec	0	0.544	0.147	0.437	0.021	1.149	2 min...	0.729	0.039	
20.20	39.39	16	0	0	1.000	256.000	8.000	0	18 min, 33 sec	0	0.542	0.147	0.460	0.018	1.167	3 min...	0.766	0.035	
21.21	40.40	13	0	0	2.000	384.000	16.000	0	16 min, 07 sec	0	0.550	0.147	0.443	0.028	1.168	2 min...	0.739	0.053	
22.22	41.41	16	0	0	1.000	256.000	8.000	0	18 min, 00 sec	0	0.501	0.148	0.432	0.011	1.092	3 min...	0.720	0.020	
23.23	42.42	14	0	0	2.000	384.000	16.000	0	16 min, 13 sec	0	0.558	0.147	0.468	0.025	1.198	3 min...	0.780	0.047	
24.24	43.43	14	0	0	2.000	384.000	16.000	0	15 min, 35 sec	0	0.520	0.148	0.451	0.024	1.144	3 min...	0.752	0.046	
25.25	44.44	16	0	0	1.000	256.000	8.000	0	19 min, 39 sec	0	0.443	0.150	0.412	0.021	1.026	3 min...	0.686	0.039	
26.26	45.45	12	0	0	3.000	512.000	24.000	0	14 min, 33 sec	0	0.429	0.151	0.450	0.034	1.063	2 min...	0.750	0.063	
27.27	46.46	13	0	0	2.000	384.000	16.000	0	16 min, 07 sec	0	0.546	0.147	0.437	0.017	1.148	2 min...	0.729	0.033	
28.28	47.47	15	0	0	2.000	384.000	16.000	0	15 min, 43 sec	0	0.514	0.148	0.440	0.027	1.129	3 min...	0.733	0.052	
29.29	48.48	16	0	0	1.000	256.000	8.000	0	19 min, 39 sec	0	0.460	0.150	0.415	0.036	1.060	3 min...	0.692	0.068	
30.30	49.49	16	0	0	1.000	256.000	8.000	0	18 min, 33 sec	0	0.536	0.147	0.446	0.012	1.142	3 min...	0.744	0.023	
31.31	50.50	16	0	0	1.000	256.000	8.000	0	19 min, 39 sec	0	0.522	0.148	0.595	0.019	1.284	3 min...	0.992	0.036	
Avg	14.673	0.000	0.041	1.694	344.816	13.000	0.000	16 min, 47 sec	0.000	0.508	0.148	0.463	0.030	1.150	3 min...	0.772	0.057		

Figure 4.4.2: Collect view plugin.

The radio logger tool of Cooja generated pcap files. Several tools are available for analyzing and interpreting network behaviors. The Wireshark was used in this study to analyze pcap files. Wireshark is one of the most popular and widely used tools for displaying IPv6 files. Fig 4.4.3 depicts the capture of Wireshark which includes the set of messages exchanged between the nodes such as ICPMv6 control messages, IEE802.15.4, and other control packets.

No.	Time	Source	Destination	Protocol	Length	Info
14627	778.584000	fe80::212:740d:d:d0d	fe80::212:7435:35:353	ICMPv6	78	RPL Control (Destination Advertisement Object)
14628	778.584000	fe80::212:740d:d:d0d	fe80::212:7435:35:353	ICMPv6	78	RPL Control (Destination Advertisement Object)
14629	779.681000			IEEE 802	7	Ack
14630	779.728000	00:12:74:1f:00:1f:1f:	00:12:74:01:00:01:01:	6LoWPAN	106	Data, Dst: NitLab_01:00:01:01:01, Src: NitLab_1f:00:1f:1f:1f
14631	779.731000			IEEE 802	7	Ack
14632	780.375000	fe80::212:7431:31:313	fe80::212:7420:20:202	ICMPv6	78	RPL Control (Destination Advertisement Object)
14633	779.473000	fe80::212:7431:31:313	fe80::212:7420:20:202	ICMPv6	78	RPL Control (Destination Advertisement Object)
14634	779.525000	fe80::212:7431:31:313	fe80::212:7420:20:202	ICMPv6	78	RPL Control (Destination Advertisement Object)
14635	779.526000	fe80::212:7431:31:313	fe80::212:7420:20:202	ICMPv6	78	RPL Control (Destination Advertisement Object)
14636	779.526000	fe80::212:7431:31:313	fe80::212:7420:20:202	ICMPv6	78	RPL Control (Destination Advertisement Object)
14637	779.526000	2002:db8::212:7416:16	2002:db8::1	UDP	89	Source port: 8775 Destination port: ggz
14638	779.527000	fe80::212:7431:31:313	fe80::212:7420:20:202	ICMPv6	78	RPL Control (Destination Advertisement Object)
14639	779.527000	2002:db8::212:7416:16	2002:db8::1	UDP	89	Source port: 8775 Destination port: ggz

Figure 4.4.3: Wireshark packet analyzer

## Simulation setup

The performance of RPL-based load-balancing routing algorithms has been evaluated on the Cooja network simulator which is a java-based simulator. The effectiveness of routing algorithms in a network was studied through simulations of the implemented algorithms. The simulator utilizes the Linux-based Contiki OS, which was made specifically for wireless sensor networks. Simulations of the algorithms were run to learn how they operate on IoT networks with respect to metrics like packet delivery ratio, power consumption, end-to-end delay, total routing metrics utilized in the network (RT-Metrics) and control overhead. One hundred (100) nodes were dispersed throughout a 100x100 meter grid, with a single root node serving as the network's central hub. The algorithms were simulated individually for twenty minutes in each simulation. The simulations were carried out in an incrementing network density from ten (10) to one hundred (100) nodes to estimate the behavior load balancing routing algorithms. The skymote (Tmote sky) was used for node simulation whose firmware is embedded in Contiki. The radio transceiver in a Tmote sky mote is powered by an MSP430 microprocessor and conforms to the IEEE802.15.4 standard. In this investigation, we use a distance-loss model for the radio medium based on the unit disk graph. During the simulation, each node generated 20 packets per minute since IoT device in the network area usually generates a large number of packets. Table 4.4.1 defines the settings that controlled the simulations in the network.

*Table 4.4.1: Parameters for simulation*

Parameters	Value
Operating System	Contiki 2.7
Propagation Model	UDGM Distance loss
Hardware platform	Skymote
Data traffic type	Point-to-point, Point-to-Multipoint, and Multipoint-to-Point
Topology	Random
Routing Algorithms	BLRPL, ECLRPL, EN-ETXPC-RPL, RPL
Size of network	10,20,30,40,50,60,70,80,90,100 nodes
Simulation Time	20 Minutes
Simulation square area	100X100 m

## 4.5 Evaluation

The performance evaluation of the RPL-based load balancing was conducted after collecting the results of every simulation. We evaluated the performance of the RPL-based load-balancing routing algorithms based on reliability and stability performance metrics to identify the effective RPL-based load-balancing routing algorithm for IoT networks. Evaluation criteria based on measures of reliability including packet delivery ratio (PDR), network delay, and power consumption while reliability metrics considered in this study are control overhead and routes utilized to transmit data.

### Performance metrics

**Packet delivery ratio (PDR)** The packet delivery ratio is the ratio between the number of packets received at the root node and the total number of packets sent by the source node throughout the simulation. The quality of the links between nodes determines which packets reach their destination. The higher the link quality, the higher the possibility of the packets being delivered to the next hop successfully. The reliability of the method in the network can be estimated from the packet delivery ratio. The algorithm that has a high packet delivery ratio is considered a reliable routing algorithm (Kalyani and Vydeki 2018). The equation (4.5.1) was used to calculate the ratio:

$$PDR = \frac{\text{totalPackets received}}{\text{total packet sent}} \times 100 \quad (4.5.1)$$

### Power consumption

The nodes deployed in IoT environments collect data, process data, and transmit the collected data. For the nodes to perform these processes they need a lot of power. IoT nodes are resource-constrained and low-powered, therefore power consumption becomes the critical performance metric. For the nodes to communicate in the network, they need to have enough power to operate for a long time. The effectiveness of the load-balancing routing algorithm will be rated based on the power consumed by the nodes. The energy consumption determines the estimated power consumed by each node while listening (LPM), receiving (Rx), data processing (CPU), and transmitting (Tx) the data. The power consumption is predicted based on the equation (4.5.2):

$$P_{Components} = P_{cpu} + P_{lpm} + P_{tx} + P_{rx} \quad (4.5.2)$$

### **End-to-End delay**

Estimating the network delay with this performance metric means knowing how long it takes for a packet to travel from its origination point to its final destination. The amount of time packets spend traveling through the network is evidence that the routing measure being used is effective. The longevity of a network improves with decreased latency between nodes. The total time taken for this analysis is described by the equation (4.5.3).

$$End - to - End \ delay = \frac{recvTime - SenTime}{total \ packtes \ Recieved} \quad (4.5.3)$$

### **Rt-Metric**

The Rt-Metric routing metric calculates the paths taken by each node in the network, from the leaf to the root, and back again, taking into account any retransmissions. Retransmission of data is reduced when the leaf node chooses a trustworthy path and parent node, leading to a lower rt-metric value. When the leaf node tried to transit packets to the parent node and the packets are rejected, it retransmits those packets to another node using another route, therefore, there is a high retransmission rate which contributes to high rt-metric. High rt-metric may contribute to higher power consumption and network delay. This performance metric determines the stability of the algorithm to the network if the algorithm maintains the minimum, rt-metric is reliable to the network.

### **Control overhead**

The RPL network topology is constructed and maintained by DODAG, which utilizes a variety of Internet Regulate messages (ICMPv6) to control the network topology. Control messages are usually generated by the nodes when they join the network (DAO), ask to join the network (DIS) and advertise themselves as neighbor nodes after joining the network (DIO).

RPL reduces the number of control plane packets by reducing unnecessary messages using tickle timers. RPL reduces the number of control plane packets by reducing unnecessary messages using tickle timers. The network is built by the control messages before the nodes begin exchanging

packets to reach the sink node (DODAG root). The goal of this measure of performance is to examine the control overhead under both normal and abnormal network situations. Higher control overhead determines the failures that occurred in the network; therefore, the network is unstable. The control overhead is the entire sum of all types of control messages relative to the total number of data packets broadcast to the network. In his investigation, the control overhead is determined using the equation (4.5.4).

$$\text{Control Overhead} = \frac{\text{total control Packets}}{\text{total data packets}} \times 100 \quad (4.5.4)$$

In this study, the aforementioned performance indicators are employed to assess the routing algorithms' performance in IoT networks with varying network densities. They provide the difference that makes a successful evaluation.

## 4.6 Communication

The research method used to design the artifacts is experimentation. The RPL-based load-balancing routing algorithms were implemented in the Contiki operating system. They were developed with different factors of routing such as routing metrics and evaluated using the same performance metrics on the same network densities. Publications from this study include the publication of the literature review which was published at the IMITEC 2019 conference.

This study started by finding the problem with the existing RPL load-balancing routing algorithms. The problem identification survey was presented at the IMITEC conference in 2019 to communicate the current problem we were facing with other scholars.

The existing RPL-based load-balancing routing algorithms were reviewed and it was found that some of them were praised for being the best performer in certain network scenarios but did not perform effectively in other network scenarios like ETXPC-RPL(Altwassi *et al.*, 2019). The ETXPC-RPL outperformed RPL in burst traffic scenarios and improved network lifetime, however, the algorithm was reliable in a limited network of 30 nodes only. We designed an enhancement for EXTPC-RPL to improve its performance in IoT network scenarios. The enhancement's performance was demonstrated in a simulation environment of an IoT network ranging from 10 nodes network up to one hundred nodes (100). The enhancement was presented

at the IMITEC 2020 Conference. We designed another RPL-based load-balancing routing algorithm enhancement named Enhanced Context aware and load balancing routing algorithm for RPL (ECLRPL). We evaluated the performance of the algorithm in IoT networks of small and large networks based on different performance metrics. The algorithm was presented at the IMITEC 2021 to communicate the importance of combining the relevant load-balancing routing metrics when routing in IoT network scenarios.

### **Contribution**

This study contributes to the body of knowledge by introducing a common load-balancing routing metric in all developed RPL-based load-balancing routing algorithms which is buffer occupancy. The proposed RPL-based load-balancing routing algorithms were evaluated in the same network scenario to understand how each algorithm performs compared to the other. The performance based on common performance metrics was evaluated to understand which load-balancing routing algorithm is a reliable and stabilizing network in IoT. The contributions of this study are valuable to the IETF group and the scholars in WSNs, IoT, and Routing over low power and lossy network work group.

The scholars with enough equipment can implement these algorithms and test them in the test bed scenario since this study is limited to a simulation environment. The limitations of this study are further explained in the following section.

## **4.7 Limitations of the research methods**

In this study, we chose the experimentation method to develop the designed algorithms. To demonstrate the implemented algorithms, we chose simulation because the study was limited to a two-year timeline, and the authors to not have enough time to deploy all the algorithms in the testbed and evaluate their performance. Therefore, the simulation was the best method to deploy the network and evaluate the performance of the algorithms one by one.

## **4.8 Summary**

This study aims to find effective RPL-based load-balancing routing algorithms for IoT networks. The design science research method (DSRM) was used to conduct this investigation. The DSRM

is the process of conducting the study based on different steps. In the first stage, the researcher must define the problem at hand and provide an argument for why it is crucial to find a solution. The gap was discovered through a review of the literature. At the beginning of the research, reviewing the current research papers, and journal articles based on the Internet of things, Routing protocols for Low power and lossy networks (RPL) were done thoroughly to master the state of the art in IoT routing protocols which is a literature review process. A literature review is a study of scholarly sources based on research topics. It provides a summary, and critical evaluation of all available works related to the research and clarifies the gaps (Duke 2002). The literature review helped to understand the RPL protocol, and its performance in different network scenarios and understand its drawbacks. The literature revealed that RPL underperforms in heterogeneous and complex IoT networks because it does not balance the load distribution effectively. However, several authors designed different RPL-based load-balancing routing algorithms(Altwassi et al 2019; Babulak 2020; Kumar and Hariharan 2020; Sebastian 2020) which balanced the load distribution in different network scenarios and have drawbacks when deployed on another network. From the literature, we selected the RPL-based load-balancing routing algorithms which outperformed in a critical network scenario such as a heavy IoT industry and used them in this study to find the effective and reliable RPL load-balancing routing algorithm for IoT networks. The related work from previous studies was extracted from different science libraries such as Google Scholar, Science Direct, IEEE, Springer, Scopus, etc. The detailed literature review is found in the previous chapter.

The second step of DSRM is to formulate the objectives of the study. To achieve every goal, there should be objectives that drive the study. The third step is to design and develop the artifact mentioned in the objectives to solve the identified problem. This study designed the RPL-based load-balancing routing algorithms and implemented them in the Contiki operating system. The research was based on a desktop. However, the experiments were conducted on a 1.8 GHz Intel (R) core i7 with 16 GB RAM. The VMWare workstation player 16.0 was installed with Contiki OS 2.7.

The fourth step of DSRM is to demonstrate the developed artifact. In this study, we demonstrated the designed RPL-based load balancing routing algorithm's performance in a simulation environment because of time frames. The demonstration was time-consuming because every algorithm was evaluated from a minimum of ten (10) nodes up to one hundred (100) nodes network

density. The Cooja network simulator was utilized to demonstrate the performance of RPL-based load-balancing algorithms in IoT networks. The simulation of the algorithms was conducted to examine the performance of the load-balancing routing algorithms in an IoT network. The simulations assisted in finding the effective RPL-based load-balancing routing algorithm for IoT networks which is the answer to the second research question. Simulations were chosen because they mimic the physical deployment of the devices and provide reliable results. The Cooja network simulator has the advantages of supporting new changes in the routing protocols design, saving simulations of the routing algorithms and comparing the performance later, and re-running the experiment. It is distributed freely; open source and it is popularly used in academic institutions and research industries.

The fifth step of DSRM is to evaluate the performance of the designed artifact. We installed Anaconda distribution for python and R programming to visualize the data. The data collected after simulations was stored in an Excel spreadsheet and visualized in Python 3.2. In this study, the performance of the routing algorithms was evaluated based on different performance metrics such as packet delivery ratio, power consumption, end-to-end delay, control overhead, and RT-Metric. These metrics were used to identify the effective RPL-based load-balancing routing algorithm in an IoT network based on its reliability and stability in the network.

The last step of this process is the communication of the study. The scholarly publications were presented at the IMITEC conference from 2019 to 2021 and in the Journal Of theoretical and Applied Information Technology.

Thorough evaluations conducted produced results that were analyzed to find the best-performing load-balancing routing algorithm for RPL in IoT networks. Python API was used for data analysis. The result of the analyses is presented in chapter 5. The graphical results are presented to show the behavior of load-balancing routing algorithms in IoT networks. Conclusions were drawn up from analyzing the graphical result to recommend the potential load-balancing routing algorithm for the future and which routing metrics can be recommended for future RPL load-balancing algorithms in IoT networks.

# Chapter 5

## Results and Discussion

Evaluation is the process of examining the performance of the load-balancing routing algorithms while exchanging data for a certain time in the network. The simulation results obtained in Chapter four are evaluated in this chapter, to find the performance of RPL-Based load-balancing routing algorithms in IoT networks. This section aims to address the second research question: **how does the RPL-based load-balancing routing algorithms perform in IoT network?**

This study aims to find an effective RPL-based load-balancing routing algorithm for IoT networks. To achieve this goal, we implemented three RPL-based load balancing routing algorithms namely ECLRPL, BLRPL, and Enhanced ETXPC-RPL in Contiki OS. We simulated the algorithms in the Cooja network simulator to find the performance of the algorithms in the growing network. The results obtained from each simulation were saved on an Excel spreadsheet and visualized in Python.

### 5.1 Results

This sub section presents the results gained from the simulation of RPL-based load-balancing routing algorithms in order to determine the efficient RPL load-balancing routing algorithm for Internet of Things (IoT) networks. The results are based on performance metrics which include packet delivery ratio, power consumption, end-to-end-delay, RT-Metric, and control overhead.

#### **Packet delivery ratio**

The packet delivery ratio is depicted in Fig 5.1.1, where we observe the relationship between the network size and packet delivery ratio. When the network size increases, there is an increase in the packets delivered to the destination. In low network sizes, there are fewer packets delivered due to the distance among the nodes. ECLRPL performs better in packet delivery ratio. In low network density, it successfully transmits almost 68% and above and in high network scenarios almost 100% of packets reached the root node. Then BLRPL becomes the second-best performer which transmits 64 % in low network density and 96% in large network density and EN-ETXPC-RPL is

the worst-performer which transmits a minimum of 56% in low network density and a maximum of 93% in large network densities in packet delivery ratio.

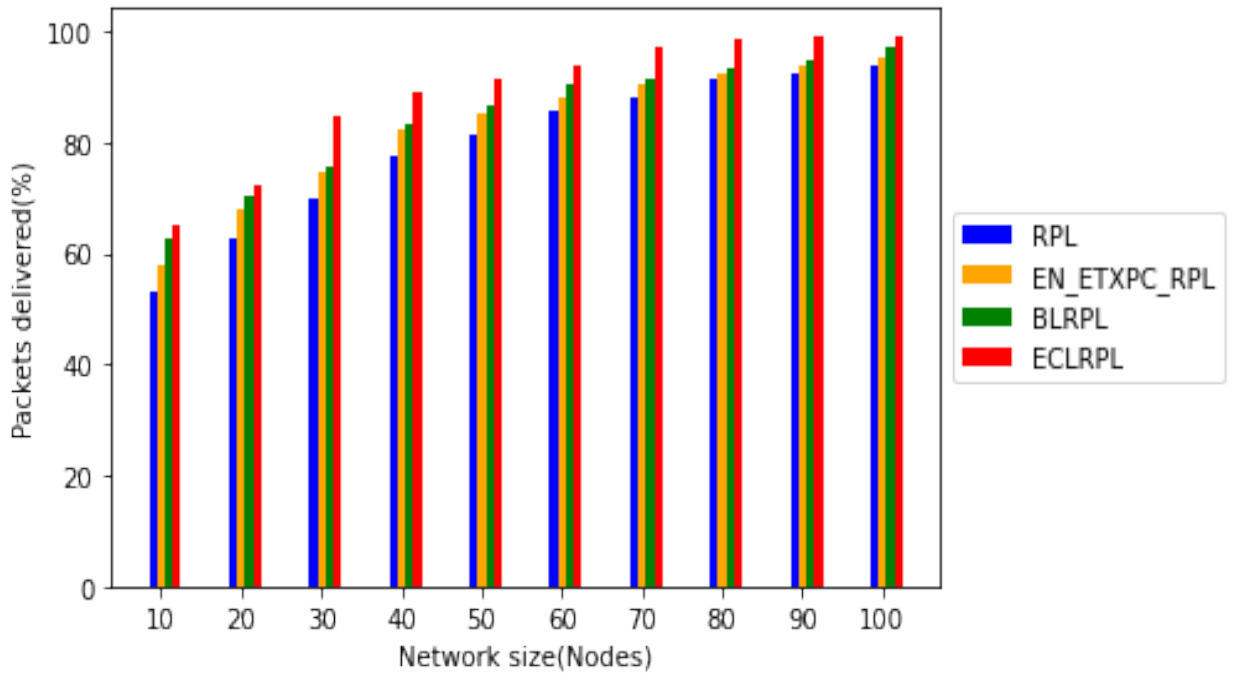


Figure 5.1.1: PDR of RPL-based load balancing routing algorithms in IoT network

### Power consumption

The power consumed by the nodes in the network comprises different tasks that occurred in the network. The nodes use power to transmit packets, receive packets, process the packet, and listen to incoming packets from different paths.

### CPU Power

The power is consumed to process the data packet or routing formation or find a reliable route to transmit data. In fig .5.1.2, it is observed that, as the network size increases, the processing power also increases. However, BLRPL has consumed low power in processing because it has few components to consider while routing. The ECLRPL shows high CPU power than BLRPL because it has more components to consider when choosing the parent node and routing path. EN-ETXPC-RPL consumes more power than other enhanced load-balancing routing algorithms in the network because the routing path procedures are inefficient. Then more power is consumed looking for an effective routing path and parent nodes.

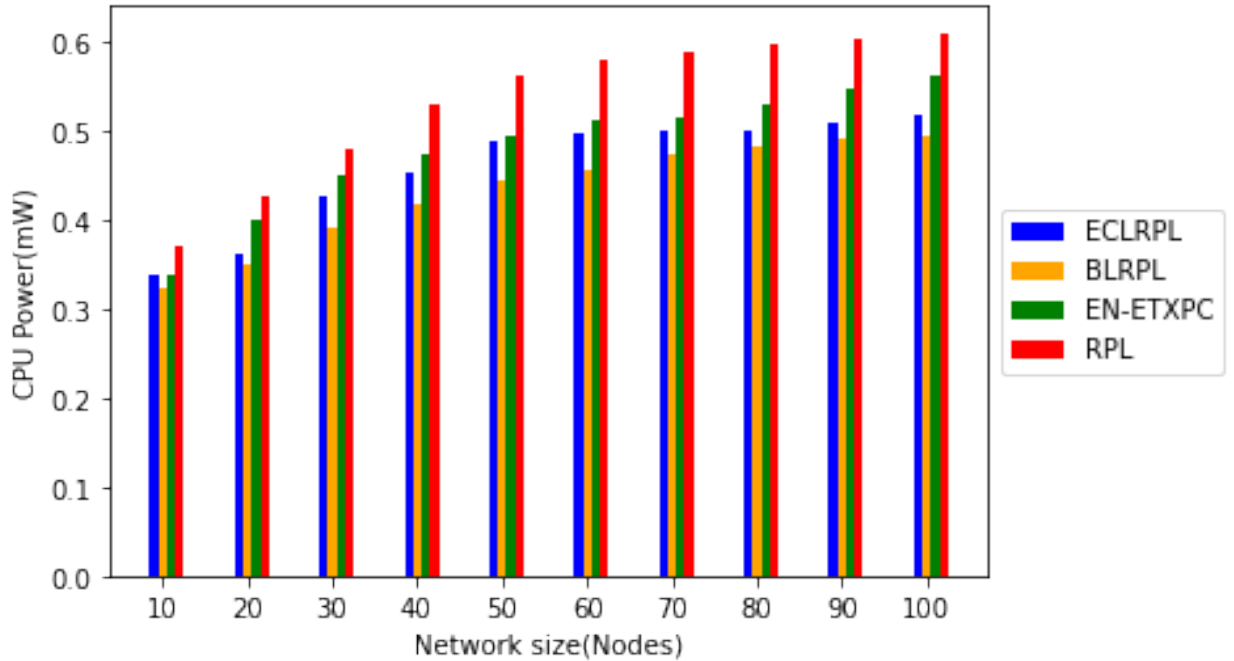


Figure 5.1.2: Power consumed by the controller in the network.

### LPM power

The LPM power is the power consumed when the nodes are in sleep mode. The LPM power decreases with the increase in the network density because when the number of nodes increases, there are many alternative parent nodes to transmit data. In large network density, the nodes are closer to each other, and they have chances to transmit data effectively and sleep to wait for incoming packets. In Fig 5.1.3, we noticed that BLRPL has higher LPM power than other load-balancing routing algorithms in IoT networks. High LPM power means that BLRPL effectively balances the load, and the nodes have enough time to sleep after transmitting the data to another node. In the ECLRPL load balancing routing algorithm, the nodes have less time to rest due to the congestion avoidance mechanism that keeps the nodes busy all the time. Therefore, ECLRPL consumes less power in sleep mode. In EN-ETXPC-RPL the nodes relax and wake up when the control overhead rises, therefore when control overhead increases, EN-ETXPC-RPL decreases the lpm power in the network.

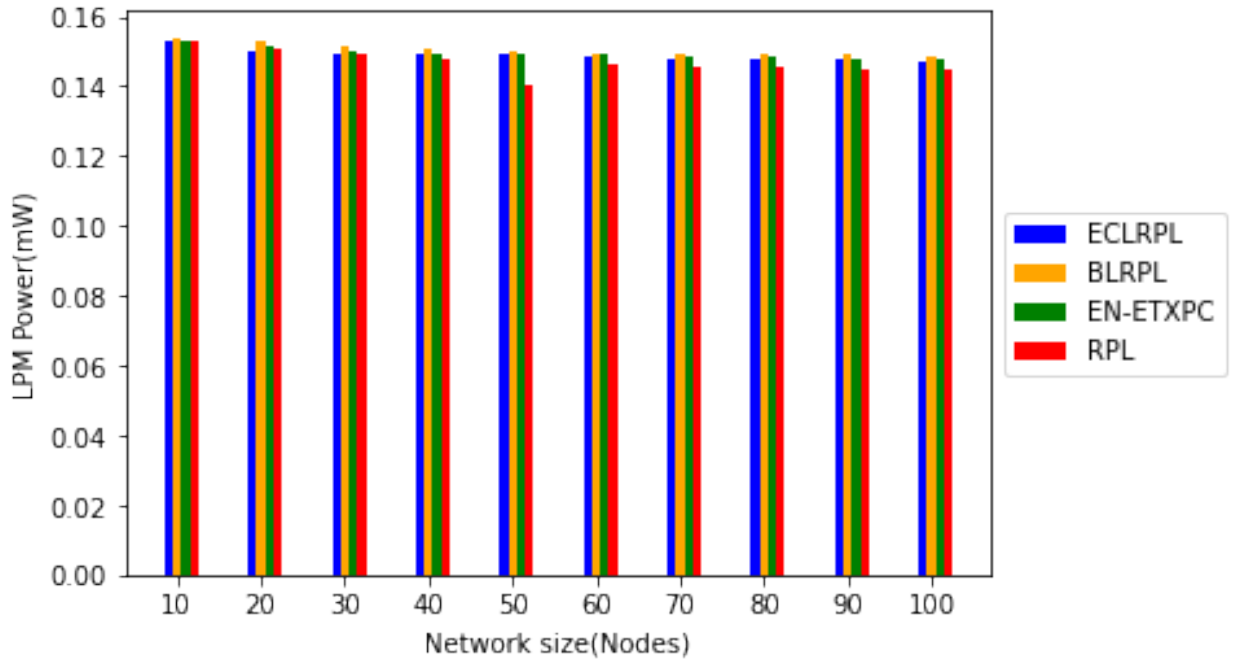


Figure 5.1.3: Power consumed in a sleep mode.

### Listening Power

Listening power is the amount of energy used by nodes to listen for potential packet transfers or control messages from the channel. Fig.5.1.4 depicts the listening power of different algorithms in IoT network. The low network traffic consists of few transmissions that result in low power consumed to listen from the channel. In large network traffic, the nodes transmit data non-stop and other nodes transmit control messages that result in high power consumed to listen to transmission media. EN-ETXPC-RPL has higher listening power than other load-balancing routing algorithms in IoT networks because the nodes must listen for incoming packets and detect errors in the form of control messages. During times of network instability, it takes more energy for nodes to monitor for incoming packets and control messages. ECLRPL increases the listening power when the nodes listen to congestion avoidance notifications and incoming packets. It is noticed that BLRPL has lower listening power than other load-balancing routing algorithms. It effectively balances the load distribution in the network and mitigates congestion and buffer overflow, thus fewer errors may be reported.

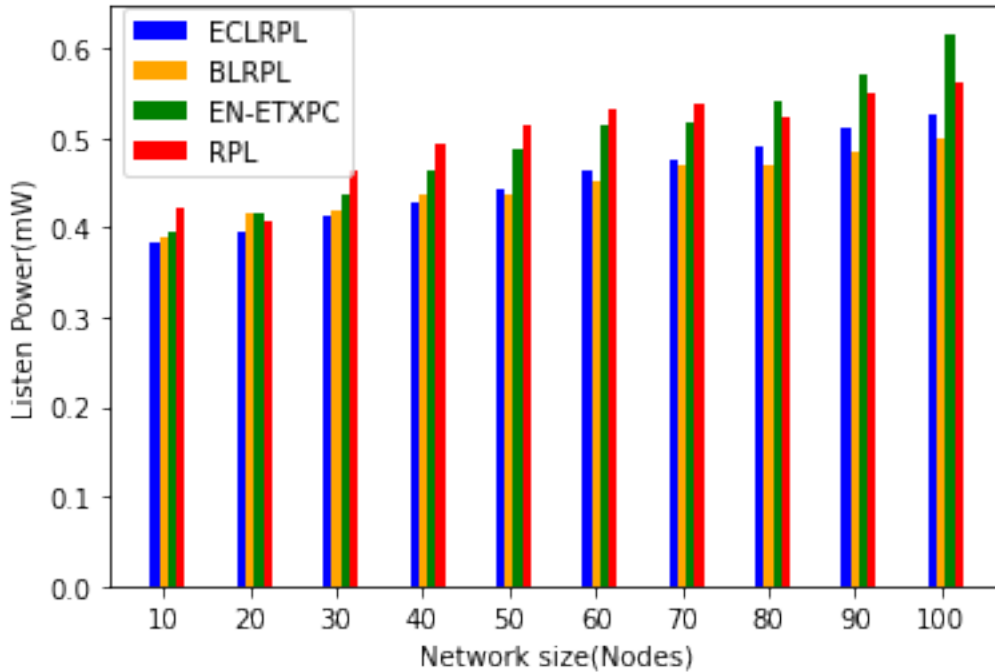


Figure 5.1.4: Power consumed to listen on the transmission medium in the network.

### Transmission power

The transmission power is the power utilized when transporting data from one location to another. It is the lowest power consumed in the network compared to the CPU power, LMP power, and listening power. The reason is that the nodes transmit the packets and sleep while waiting for a response from another node. The fig. 5.1.5 depict the transmission power of different network sizes. However, BLRPL consumes more power to transmit packets in low network traffic ranging from 10 nodes to 40 nodes compared to ECLRPL because, in the low network, traffic nodes are far from each other. The power consumed to transmit is higher due to the retransmission of data and generation of control messages for finding the reliable path and parent node. In a large network, BLRPL conserves transmission power. ECLRPL is the second-best at conserving transmission power. EN-ETXPC-RPL shows high transmission power because of the high retransmission rate caused by bounced-back packets sent to unreachable nodes. When the parent nodes are out of power they die, and the packets that were coming to that parent node are dropped or bounced back to the sender to find an alternative parent node. That process occurs in EN-ETXPC which fails to check the remaining power of the preferred parent chosen. Thus, increasing the transmit power. RPL turns out to consume more power than all routing algorithms because it

does not perform any load balancing in the network. RPL has a high chance of retransmission of power resulting from choosing poor routes and ineffective parent nodes.

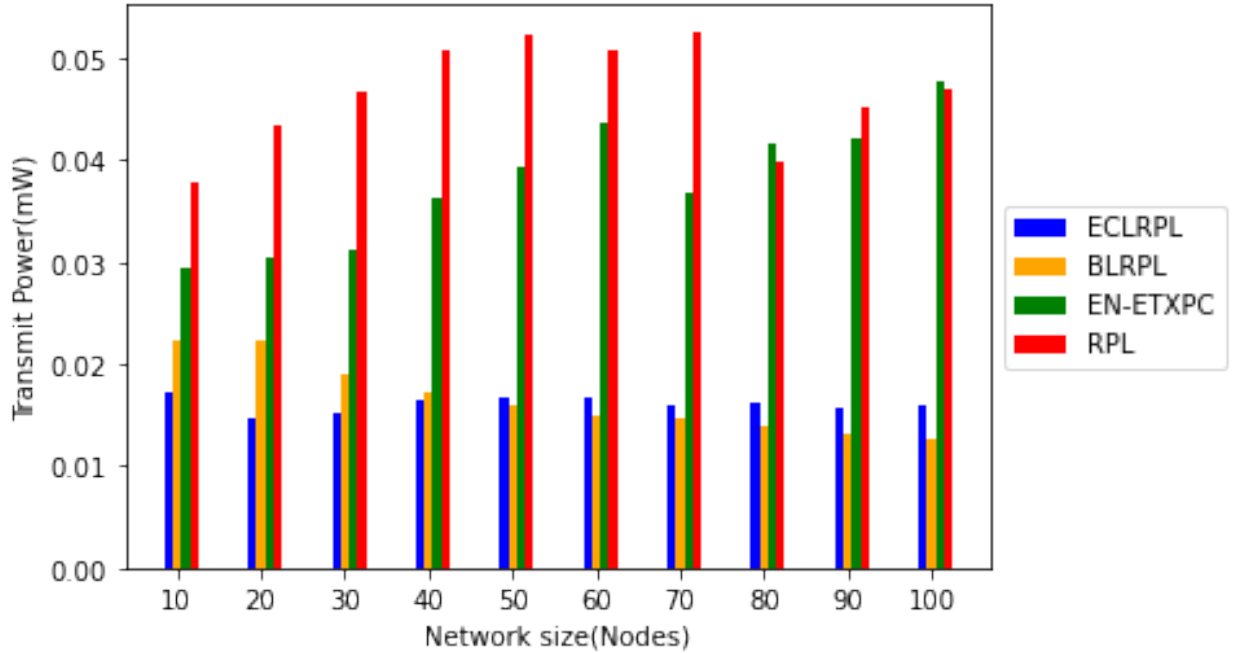


Figure 5.1.5: Transmission power of Enhanced load balancing based on RPL.

### Total Power consumption

Power consumption of network nodes is shown in Figure 5.1.6. As the Internet of Things is comprised of low-power devices, energy efficiency is a major concern. Power consumption was found to rise in proportion to network growth because more nodes are needed to detect, listen, accumulate, and communicate data. This causes an increase in overall network energy usage. Nevertheless, BLRPL uses less energy than ECLRPL, EN-ETXPC, and the original RPL algorithm. This is because it reduces the burden on the nodes and mitigates the buffer overflow. ECLRPL is the second-best conserving power in IoT networks due to its effective balancing of the load considering the residual energy on the nodes, mitigating the congestion in the nodes, and avoiding buffer overflow. These routing metrics contribute to mitigating power consumed in the network.

EN-ETXPC-RPL consumes more power than other load-balancing routing algorithms because it neglects the node's power metric when routing. The algorithm is at risk of network congestion because the child nodes transmit packets to the parent nodes without knowing the battery of the parent node. If the parent node dies, the packets will be retransmitted to other nodes and the notifications will be forwarded back and forth which will increase control overhead in the network. The nodes consume more power to stabilize the network, thus RPL consumes more power because it has an unstable network in IoT.

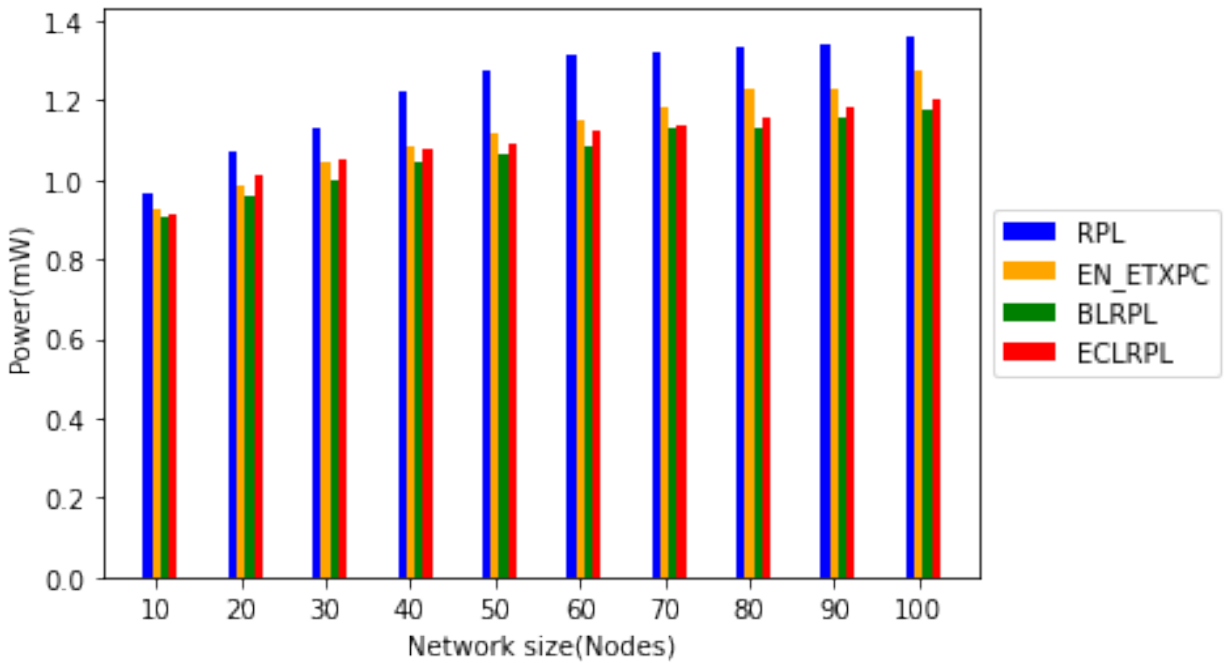


Figure 5.1.6: Power Total power Consumed during data exchange.

### Network delay

Due to its resource limitations, the LLN network generally transfers data slowly. The effective load-balancing routing algorithm tackles these problems and increases network performance. However, the end-to-end delay is one of the factors required to transmit sensor data from time to sink time. The fig 5.1.7 displays the performance of different RPL based load balancing routing algorithms in IoT network. Observations indicate that the delay rises as network size grows. This is due to the volume of data transmitted in the network needing more time to be processed. It is noted that BLRPL has a lower delay followed by ECLRPL and the EN-ETXPC-RPL in IoT networks. The network delay increases when the network is unstable, and the data packets are stacked in the queue while the network is flooded by a control message that is responsible for

finding the alternative paths and parent nodes. EN-ETXPC-RPL has a higher delay than all the load-balancing routing algorithms in IoT networks because the nodes are congested, and the channel is overloaded since it neglects the power of the node when choosing the reliable parent node. RPL does not apply any load-balancing distribution parameters therefore, it suffers from congestion, packet loss, and delay in the IoT network.

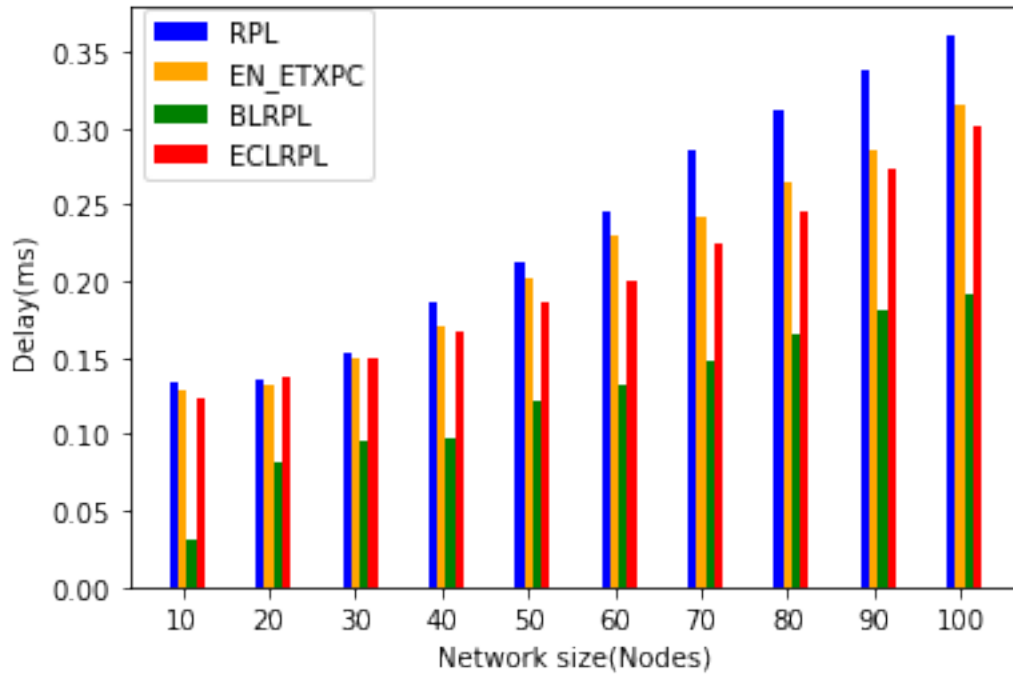


Figure.5.1.7: Delay in the network.

### Routes utilized to transmit data packets (Rt-Metric)

Each load-balancing routing algorithm uses different strategies to transmit packets, therefore, they will be differently routed to be used for data transmission. Figure 5.1.8 presents the total routes used for data transmission in different network densities. However, the more nodes are deployed on the sensing site, the more packets are generated, and they use more routes to transmit data to the root node. ECLRPL shows high routes used for data transmission. BLRPL mitigates the routes used for data transmission in IoT networks and performs better than all other enhanced load-balancing routing algorithms. EN-ETXPC-RPL becomes the second-best in reducing the routes used for data transmission.

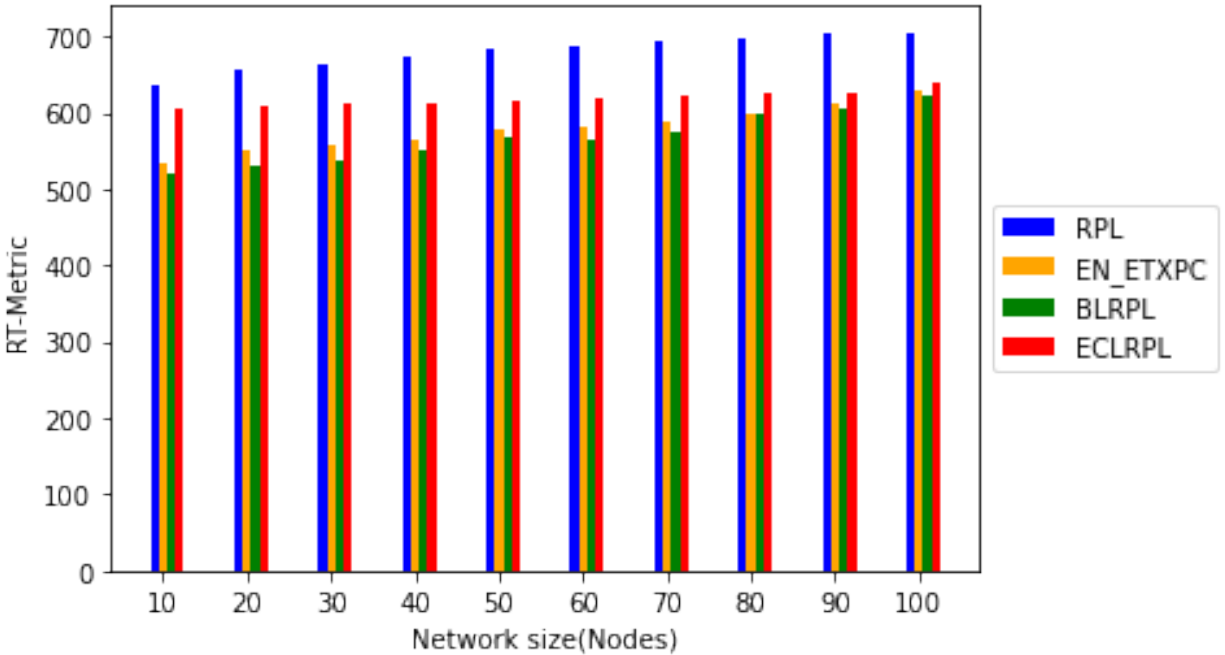


Figure 5.1.8: Total routes used for communication.

### Control Overhead

The routing topology is maintained by control messages, that are represented as control overhead. The nodes communicate well with control messages which they use to ask permission to join the network and to report the availability of the node and when the packet is transmitted. The control messages have a bad influence on the network. Figure 5.1.9 presents the control messages generated in the network. The results in Figure 5.1.9 depict that ECLRPL generates high control messages followed by EN-ETXPC-RPL. BLRPL performs excellently in mitigating the control overhead. ECLRPL is an effective load-balancing routing algorithm that performs routing considering the congestion avoidance routing metric. The congestion avoidance routing metric is responsible for reporting any matter that occurs in the network, it controls the network. Instability in the network causes an increase in the number of control messages sent, which in turn increases the network's control overhead. Therefore, ECLRPL is unstable in the IoT network due to high control messages generated throughout the network. It balances the load distribution by avoiding congestion which results in a high generation of control messages.

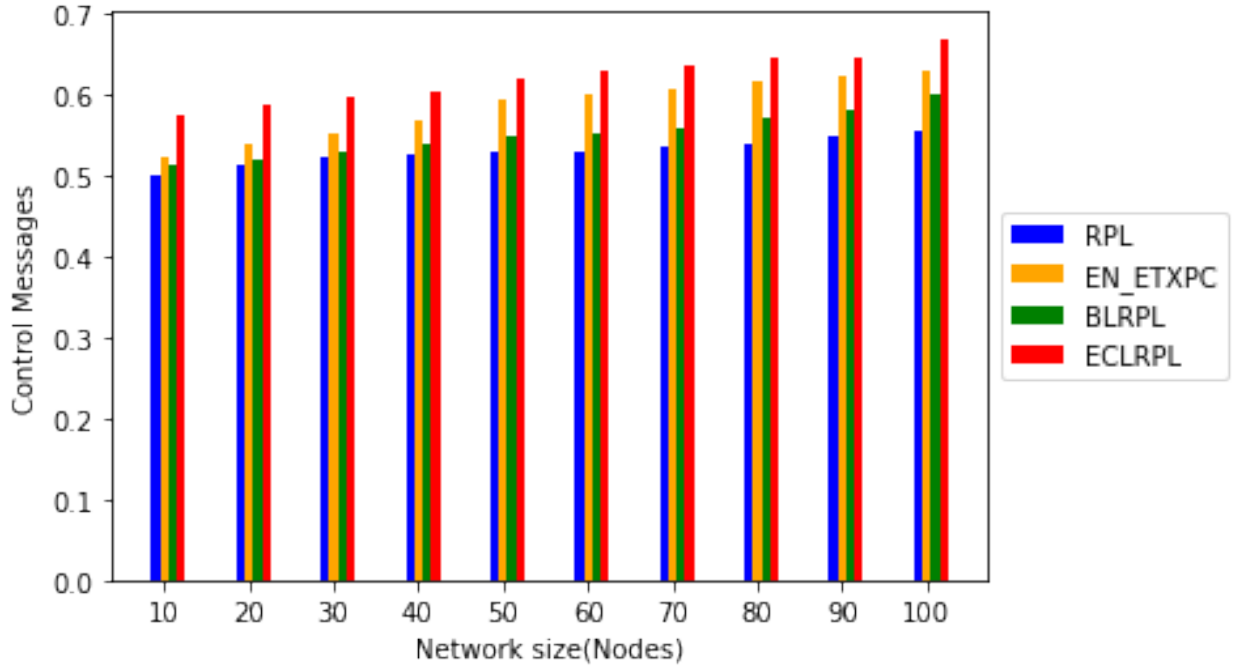


Figure 5.1.9: Control overhead of RPL-load balancing algorithms.

## 5.2 Discussion

The reliability of the RPL-based load-balancing routing algorithms to the network is evaluated based on the rate of packets delivered to the destination and the average delay of a packet traveling from the source to the destination node. As a rule of thumb, improved reliability is accompanied by increased energy consumption, typically in the form of retransmissions or acknowledgments. Consequently, achieving a balance may also increase network reliability.

ECLRPL outperforms BLRPL and EN-ETXPC in the ratio of packets delivered to the destination node: It is the second-best in reducing power consumption and end-to-end delay in an IoT network. It is the least performer in RT-Metrics and it controls overhead. The congestion avoidance mechanism enables the nodes to generate more acknowledgments during data transmission which results in high control overhead. The nodes receive the notifications that the parent node they intended to use is fully occupied and that they need to change the parent node and start to look for alternative parent nodes. Therefore, there are high routes used to transmit packets and delay the packet from reaching the destination. The performance results show that ECLRPL is unstable in

the IoT network because it generates more control messages to notify the congestion in different nodes. High control messages lead to higher utilization of routes to transmit data packets and more power is consumed by the nodes looking for alternative parent nodes and routes. When the nodes consume more power during data transmission it means that the algorithm is unreliable to the network because it is not energy efficient.

BLRPL delivers a maximum of 96% of packets successfully in Figure 5.1.1, which is lower than the packets delivered by ECLRPL but higher than packets delivered by EN-ETXPC-RPL and the default RPL in the IoT network. The algorithm mitigated network delay, power consumption, and routes used for packet transmission (RT-metric). The lesser routes used, result in lower power consumption and is fast in transmitting the packets successfully. The algorithm reduced the control overhead more than the other load-balancing routing algorithms since this algorithm enables the child nodes to transmit packets using reliable paths and select the reliable parent nodes. According to the performance results obtained above, BLRPL is reliable and stable in the IoT network. BLRPL balanced the load distribution effectively by considering the best route with the least queue and the reliable parent node with a high remaining buffer capacity and enough energy for data transmission. BLRPL is energy efficient in IoT networks.

The performance of EN-ETXPC-RPL is poor compared to ECLRPL and BLRPL in IoT networks. This is depicted by the simulation results in the above figures. EN-ETXPC-RPL is the least performer in Packet delivery ratio, power consumption, and network delay, but the second-best in mitigating the routes used for transmitting data and reducing the control overhead than ECLRPL.

Enhanced load balancing routing algorithms performance is also affected by environmental issues such as distance among the nodes. None of the nodes have an equal distance between them due to their random deployment throughout the simulation area. In low network densities, the nodes were scattered far from each other. Depending on various performance metrics, this can have a good or bad effect on the network. Long-distance in low network densities may result in a poor packet delivery ratio because some of the packets are lost during the journey of reaching the destination node. In large network densities nodes are closer to each other they have the privilege to choose the most reliable path for data transmission. This results in poor PDR in low network density and high PDR in large network density. Power consumption increases with the increase in traffic,

therefore, as few packets reach the destination there is also low power consumption and there is a little delay. The number of routes used to transmit data in low network density is small as there are few packets transmitted and few control messages are generated.

The performance evaluation of the load balancing routing algorithms in IoT network assisted us to achieve our goal of finding the effective RPL-based load balancing routing algorithm for IoT network. From the above results, BLRPL is the best-performing load-balancing routing algorithm in IoT networks because it proves the reliability of the network by improving the packet delivery ratio, reducing network delay and power consumption, and giving stability to the network by reducing the retransmission rate and control overhead than other load balancing routing algorithms. It is the second-best in packet delivery ratio. The ETX metric has an impact on poor packet delivery ratio. ETX metric may select longer paths, which may lead to packet delay and or packet loss because it does not consider the length of the path when selecting the best path for data transmission.

The evaluated RPL-based load-balancing routing algorithms performed effectively in IoT networks. From the results obtained, it appears that BLRPL is the reliable RPL-based load-balancing algorithm that stabilizes the network by mitigating the control overhead. The buffer occupancy routing metric has a great role in mitigating the control overhead which leads to low power consumption and reduced network delay.

To prove that these algorithms performed effectively, we provide the performance comparison of the proposed algorithms over the existing RPL-based load balancing routing algorithms which do not use buffer occupancy routing metrics. The performance is based on similar performance metrics that have been used to evaluate the performance of the proposed RPL-based load-balancing routing algorithms.

### **Packet delivery ratio**

The fig.5.2.1 identifies the packet delivery ratio of the existing RPL-based load-balancing routing algorithms versus the proposed RPL-based load-balancing routing algorithms in IoT networks.

EN-ETXPC-RPL improved the packet delivery ratio compared to ETXPC-RPL in IoT networks. BLRPL underperformed in packet delivery ratio compared to lbRPL in the IoT network. ECLRPL outperformed CLRPL in the IoT network. The improvements determine the good impact of load balancing in IoT networks. In ECLRPL and EN-ETXPC-RPL, buffer occupancy played a huge role in improving the packet delivery ratio. However, BLRPL has a lower packet delivery ratio compared to other RPL-based load-balancing routing algorithms in IoT networks.

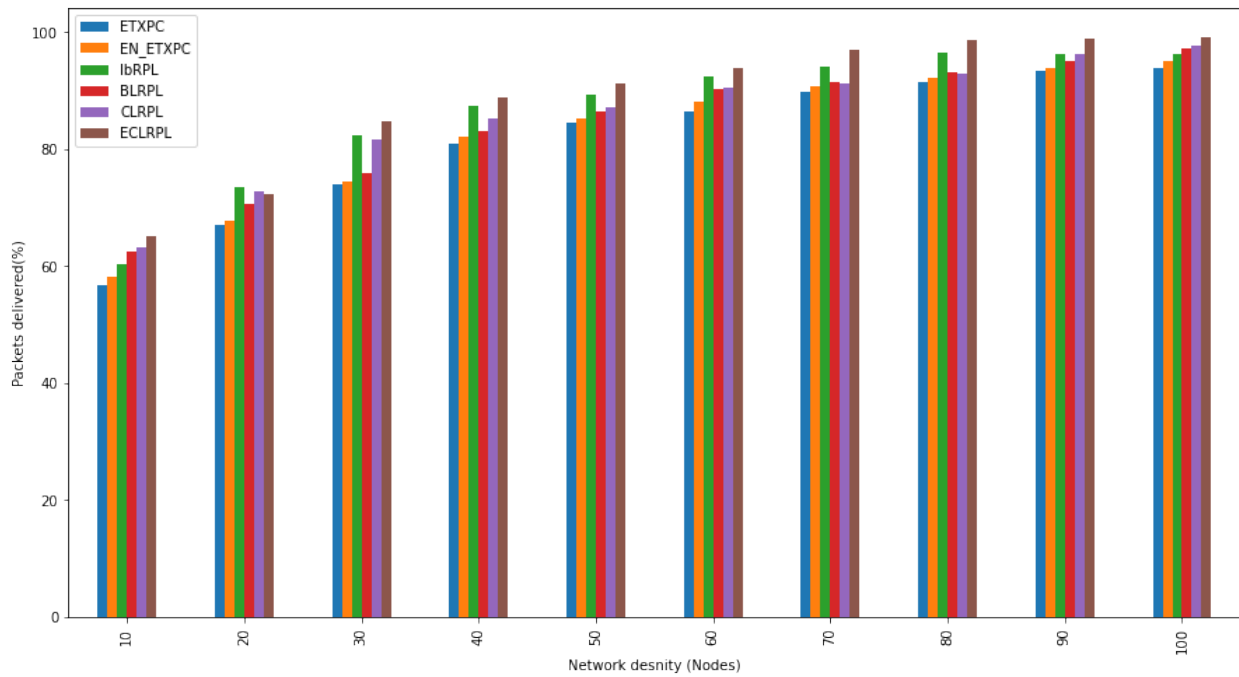


Figure 5.2.1: Packets successfully delivered to the destination in the IoT network.

### Power consumption

The nodes in the network use power to perform different duties, therefore, power consumed is categorized into four different parts namely: CPU Power, LPM Power, Listen Power, and Transmission power.

CPU Power. The fig.5.2.2 presents the CPU power consumed by different RPL based load balancing routing algorithms in IoT network.

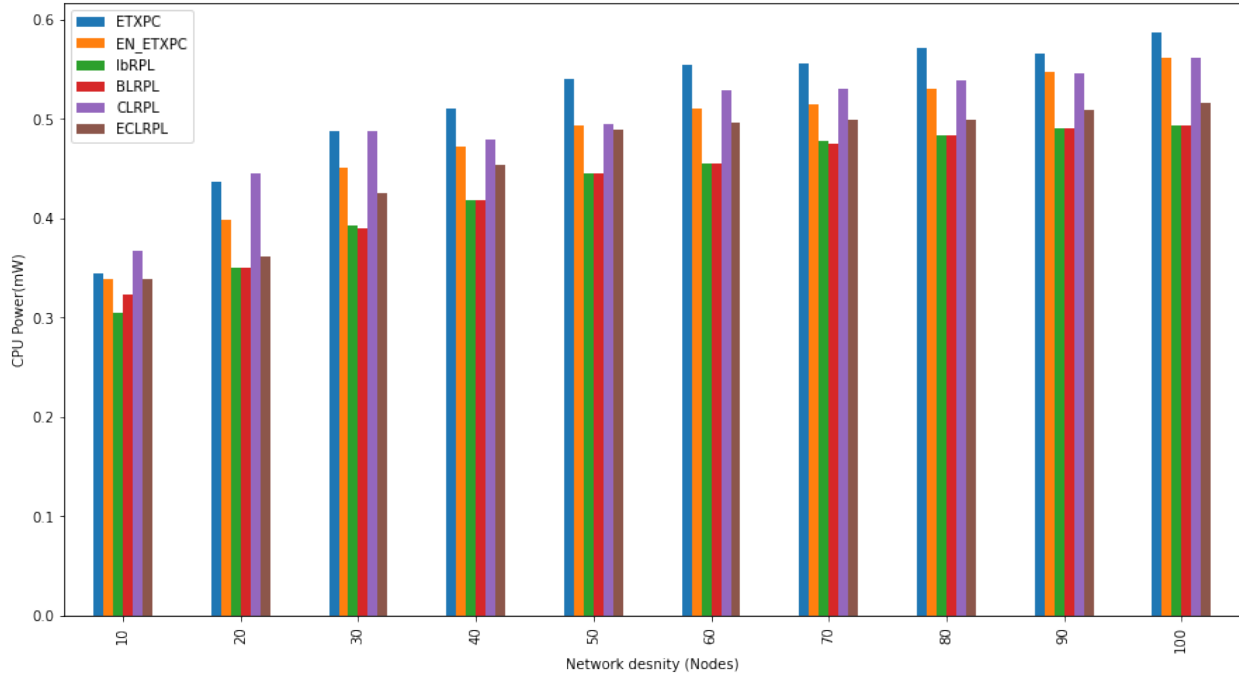


Figure 5.2.2: CPU Power in IoT network.

The observations prove that BLRPL consumes less CPU power than other RPL-based load-balancing routing algorithms. We observe that ETXPC-RPL consumed the most CPU power and it is followed by CLRPL. ECLRPL consumed less CPU power compared to EN-ETXPC-RPL and lbRPL consumed less CPU power than other RPL-based load balancing routing algorithms including the algorithms with buffer occupancy routing metric.

### LPM Power

The LPM power in fig5.2.3 identifies the power consumed by the nodes in sleep mode. The proposed RPL-based load-balancing routing algorithms stabilized the network by enabling the nodes to be in sleep mode more often. BLRPL is the algorithm that consumes more power in sleep mode; this means it balances the load distribution effectively in the IoT network and enables the nodes to stay asleep while they are not working which is a good sign of being energy efficient on the network. The buffer occupancy routing metric and remaining energy metric play a good role in BLRPL to improve network lifetime and save more energy in IoT networks while reducing the time consumed to transmit the packets.

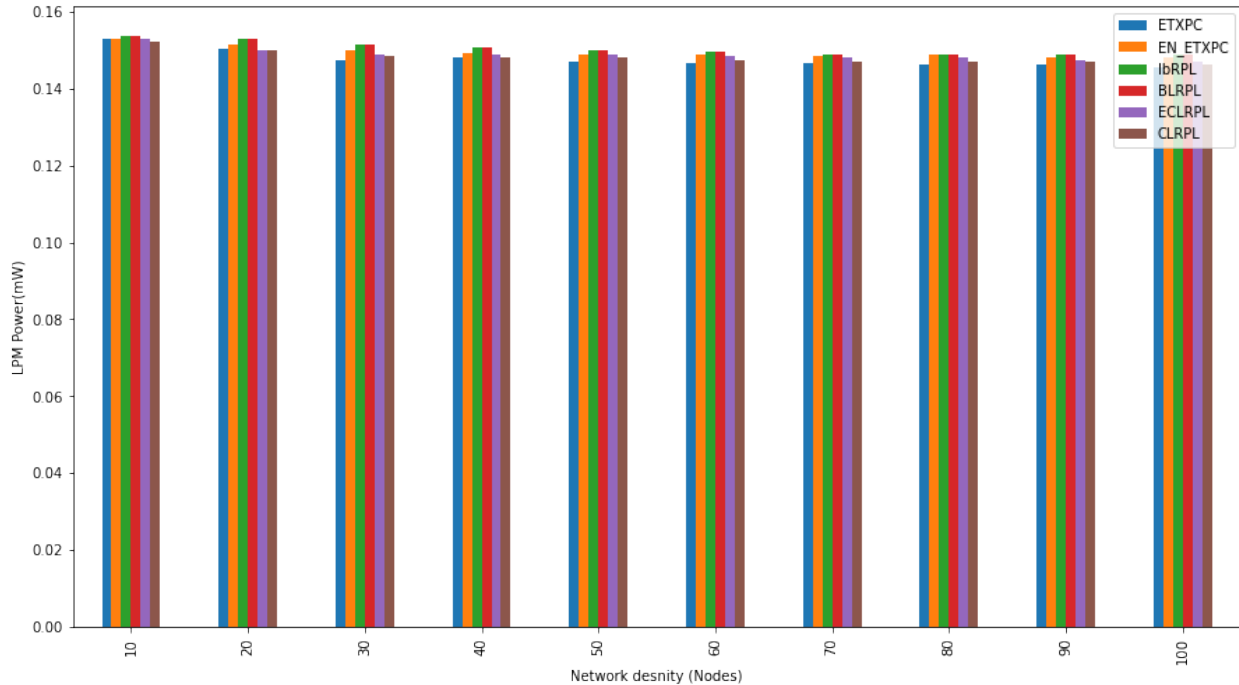


Figure 5.2.3: Power consumed while devices are in sleep mode.

### Listen power

Listen power is the power consumed to listen upcoming traffic. The proposed RPL-based load-balancing routing algorithms consume less power compared to the existing RPL-based load-balancing routing algorithms in IoT networks. This means that the proposed RPL-based load-balancing algorithms are reliable in the IoT network. The results in fig.5.2.4 depict that ETXPC-RPL consumes more power listening to the upcoming packets in low network traffic while CLRPL consumes more listening power in a high network. CLRPL consumes more power listening to the upcoming packets because it receives a lot of data from the nodes which includes the information of the packets directed to the congested nodes and bounces back. While ETX-PCRPL consumes more packets to listen to incoming packets in low network traffic because it does not consider the energy metric when it is routing. BLRPL has less listening power because when it balances the load distribution, it makes sure that all the packets are directed to the nodes with available buffer capacity; it checks the parent node's availability before transmitting the packet. Therefore, the parent nodes consume less power listening to the upcoming traffic in the IoT network.

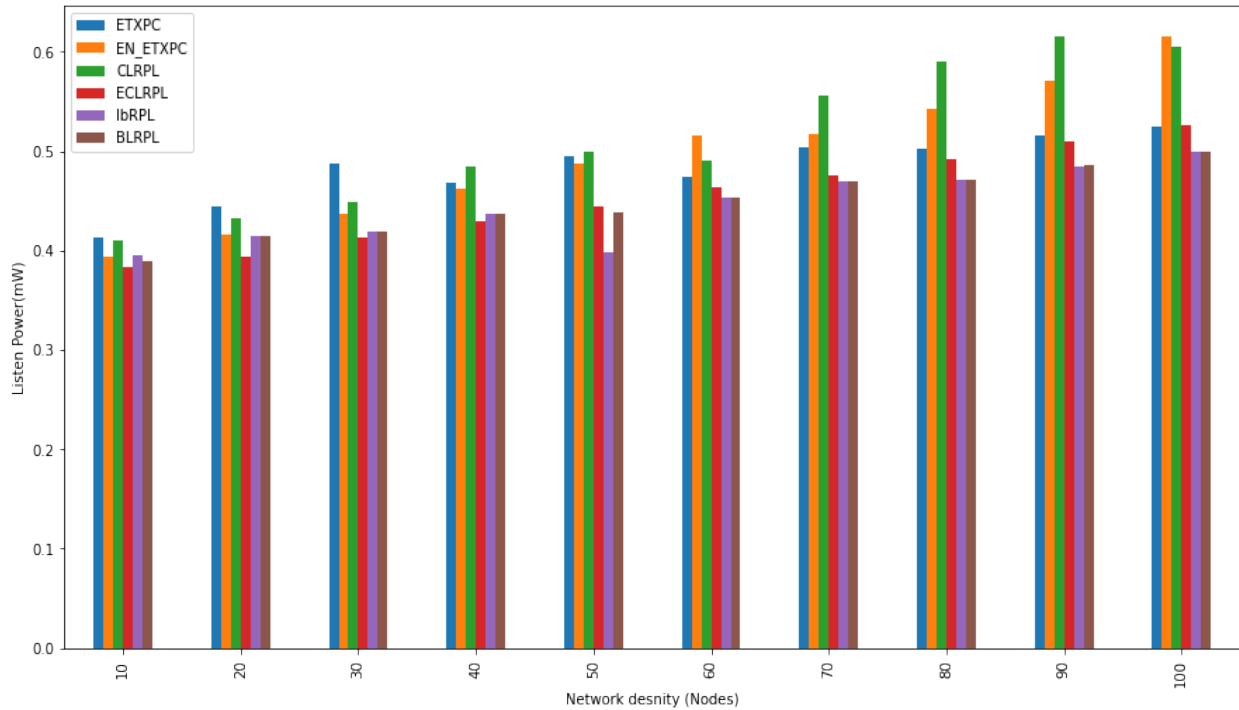


Figure 5.2.4: Power consumed to listen the incoming packets.

### Transmission power

The transmission power is depicted in fig.5.2.5. When the nodes consume power to transmit data, we noticed that CLRPL consumes more power for data transmission compared to ECLRPL. ECLRPL balanced the load effectively which assisted in mitigating the power consumption during data transmission. BLRPL conserved transmission power in large networks compared to lbRPL. EN-ETXPC-RPL consumed low power compared to ETXPC-RPL. However, BLRPL outperformed all the load-balancing routing algorithms in conserving the transmission power.

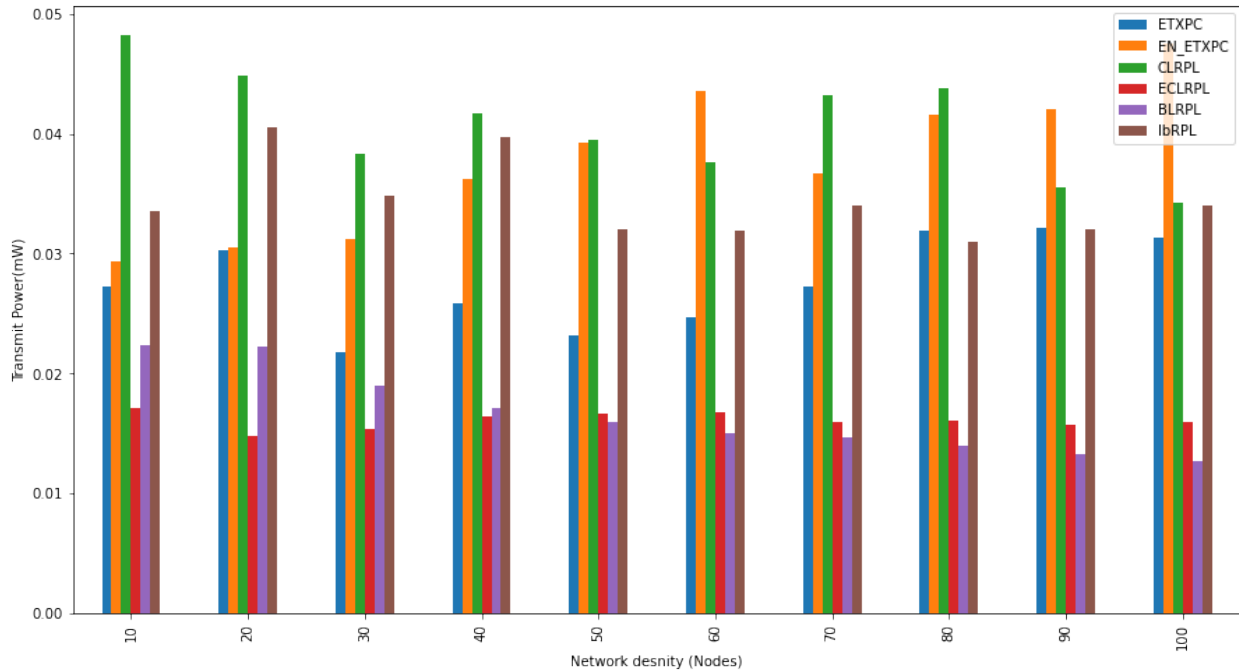


Figure 5.2.5: Power consumed to transmit data.

### Total power consumption

The total power consumption consists of the CPU power, LPM, Listen, and Transmission power. In fig.2.5.6 the total power consumption is depicted. The proposed RPL-based load balancing routing algorithms balanced the load effectively in such a way that they transmit packets with a minimum power consumed. The proposed RPL-based load-balancing routing algorithms consume low power during data transmission. The results depict that BLRPL is the most energy efficient followed by ECLRPL in the IoT network. We noticed that if the algorithm considers the energy routing metric and buffer occupancy during load balancing It lengthens the life of the network by lowering power usage and network latency, network congestion, and control overhead. Therefore, these two algorithms can be recommended for balancing the load distribution in an IoT network.

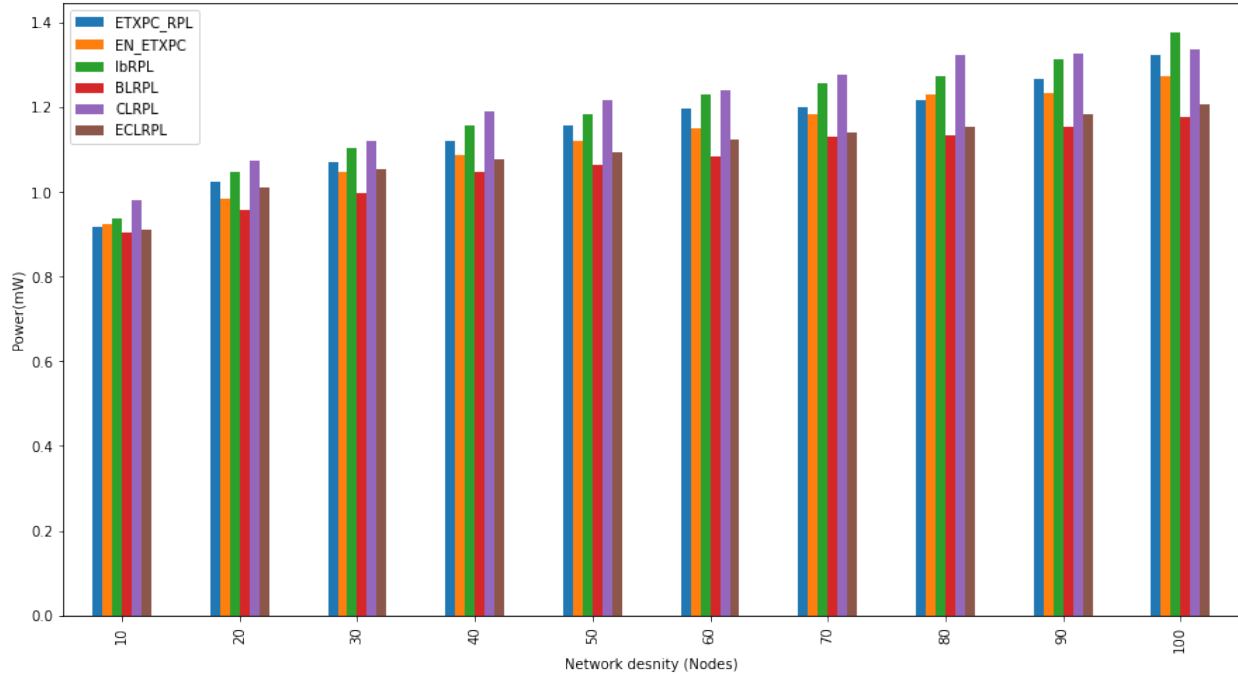


Figure 5.2.6: Total power consumed in IoT network.

## Delay

The network delay is a performance metric that identifies the time taken by the packets to reach the destination in a network. The fig5.2.7 presents the network delay of existing RPL-based load-balancing routing algorithms over the proposed RPL-based load-balancing routing algorithms in IoT networks. It is noticed that the existing RPL-based load-balancing routing algorithms delay the network while the proposed RPL-based load-balancing routing algorithms mitigate the network delay by reducing time wasted by the packets in the queues waiting to transmit in the congested nodes. BLRPL reduced network delay significantly compared to all other RPL-based load balancings.

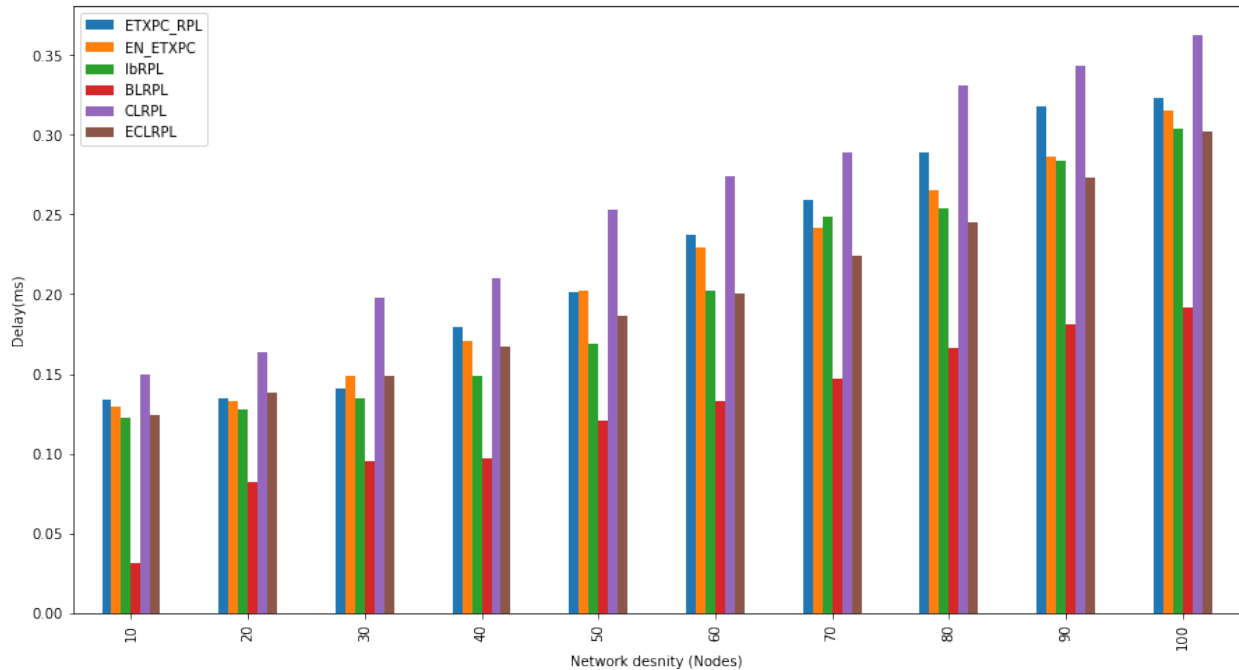


Figure 5.2.7: Network delay in different network sizes.

## RT-METRIC

The TR-Metric is the total routing metrics used by the algorithm to transmit the packets in the network, the routing metrics includes the total routes travelled by the packets to reach the destination and number of hops passed by the packets before they reach their destination. The graphical illustration of RT-Metric is depicted in fig.5.2.8. BLRPL routing algorithm used fewer routes to transmit data packets to the destination node, therefore, it became the best performer of all other RPL-based load balancing in IoT networks. EN-ETXPC-RPL utilized fewer routes to transmit packets compared to ETXPC-RPL and ECLRPL with CLRPL. This indicates that, BLRPL is able to transmit the packets with reliable paths in IoT network.

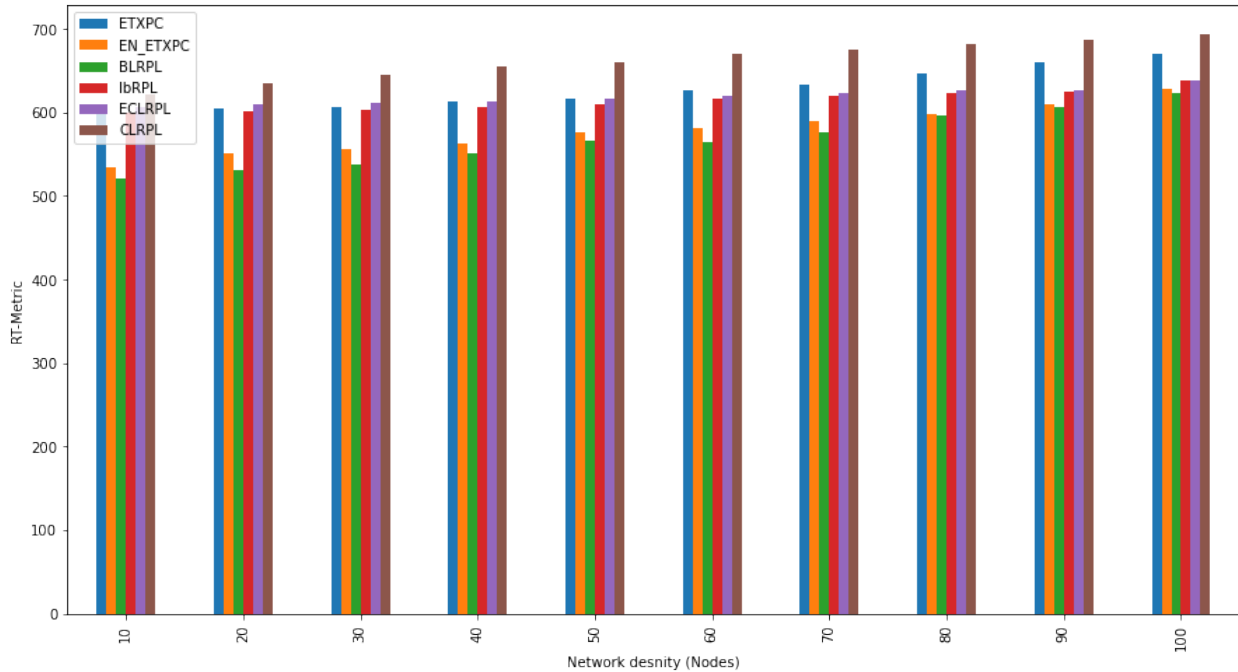


Figure 5.2.8: Routes utilized during data transmission in different network sizes.

## Control overhead

Balancing the load distribution has a positive effect on the IoT network because the proposed RPL-based load-balancing routing algorithms have low control overhead compared to the existing RPL-based network. This emphasizes the positive impact of buffer occupancy routing metrics in the algorithms. Fig.5.2.9 present the control overhead of the RPL based load balancing routing algorithms in IoT network. EN-ETXPC-RPL shows degradation of control messages compared to ETXPC-RPL. The BLRPL also mitigated control messages compared to lbRPL and ECLRPL reduced control messages compared to CLRPL. These results depict that the proposed RPL-based load-balancing routing algorithms can stabilize the network by balancing the load distribution. BLRPL is an effective load-balancing routing algorithm for IoT networks by its ability to outperform all routing algorithms in four performance metrics.

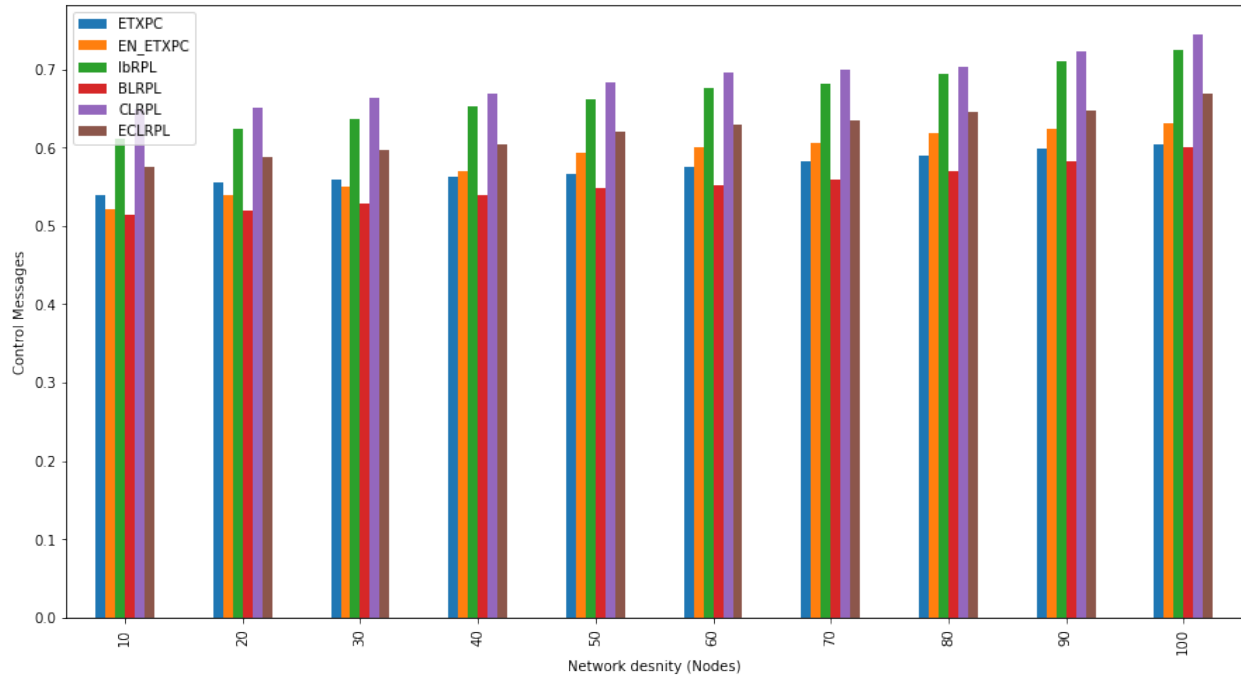


Figure 5.2.9: Control Messages generated in the network.

The aim of this study was to find the effective RPL based load balancing routing algorithm for IoT network. According to the performance of RPL-based load-balancing routing algorithms, BLRPL is the effective load-balancing routing algorithm that is reliable to the network and able to stabilize the network in IoT. BLRPL balances the load distribution with remaining energy, buffer occupancy, and ETX routing metrics. The algorithm utilized the buffer occupancy which ensures that the network is not congested, less power consumed to transmit packets in IoT network, it also used the remaining energy to ensure that the packets are sent to the parent nodes with long life span. BLRPL reduced the network delay by using ETX routing metric which enables the node to find the reliable path to transmit the packet to the parent node. Further studies can implement BLRPL in a testbed to evaluate its effectiveness in a real scenario such as the smart metering industry.

### 5.3 Recommendations

The critical evaluations of the load balancing routing algorithms results show the performance of the algorithms in IoT networks by observing the behavior based on different performance metrics. This section recommends the best-performing Enhanced load-balancing routing algorithm in IoT networks. The best-performing routing algorithm is determined by being the best performer in

three critical performance metrics namely packet delivery ratio, network delay, power consumption, and control overhead. The results in Section 5.1 show that BLRPL outperforms all load balancing in maintaining power consumption, reducing network delay, and controlling overhead significantly. BLRPL became the second-best performer in packet delivery ratio with 96% of packets transmitted to the destination. Compared to ECLRPL, it outperformed all routing algorithms with a packet delivery ratio of 99% and became the second-best in maintaining power consumption and network delay. ECLRPL performed poorly compared to all the other routing algorithms in control overhead and Rt-metric, therefore, ECLRPL is unreliable to the network due to its higher rate of retransmission of data. EN-ETXPC-RPL is the second-best on control overhead, and Rt-metric while it is the least performer in Packet delivery ratio, power consumption, and network delay. However, EN-ETXPC-RPL is the least protocol to be considered as load balancing for RPL in IoT networks. Overall, BLRPL is the best performer, and it is considered a reliable routing load balancing for IoT networks because it can transmit almost 97% of packets, maintain power consumption, and network delay and reduce the Rt-metrics and control overhead.

The evaluation results identify that BLRPL is a reliable and stable load-balancing routing algorithm for IoT networks, this algorithm can transmit packets at a minimum delay. It conserves power which is the most critical part of IoT devices since they are resource constrained. BLRPL also ensures that all the packets are transmitted through reliable routes which assisted in reducing packet loss and control overhead. This study finds BLRPL as the reliable load balancing for IoT networks due to its ability to improve the network lifetime in a large IoT network.

Buffer occupancy routing metrics significantly improved the performance of the load-balancing routing algorithms in IoT networks. The load-balancing routing algorithm applies a balanced load distribution based on different routing metrics. Adding another node metric to each load-balancing routing metric affected each load-balancing routing algorithm differently. When buffer occupancy is combined with link metric and energy-based metric, the algorithms performed better than all other load-balancing routing algorithms in 4 performance metrics. When the load-balancing routing algorithm compromises the energy-based metric, the network suffers from early nodes death and delays packets from reaching the destination that degrades the network lifetime. When the load-balancing routing algorithm compromises the best link during data transmission, the

network experiences high retransmission of data, packet loss and high network overhead. However, the suitable metrics for balancing load distribution are Buffer occupancy, ETX for the best link, and Energy-based node metrics such as residual energy or remaining energy of the node. Improved trickle algorithms should be included in the load-balancing routing metrics because they will contribute to mitigating the traffic overhead and power consumption. The single approach in RPL is recommended because it avoids routing loops and network delays. The powerful RPL load balancing for IoT networks should consider a single approach using composite metrics to select the effective parent node such as buffer occupancy, Residual Energy (or remaining energy of the node), and involve the improved trickle algorithm.

# Chapter 6

## Conclusion and Future Work

The Internet of Things (IoT) requires a reliable routing protocol for data exchange in IoT application scenarios. The Internet Engineering Task Force proposed a Routing Protocol for low-power and Lossy Networks to be the standard routing protocol for IoT networks. However, RPL has many problems in large network scenarios such as load balancing, early node death, etc. This study, therefore, describes different issues affecting the RPL protocol in different network scenarios. It also proposed solutions to improve the performance of the existing load-balancing routing algorithms in IoT networks. The contributions and the whole work that has been done in this study are concluded in this chapter which provides a summary of the methods used to achieve the goals of this study and answers the research questions that are raised in the first chapter, describes the summary of key findings as well as the future work based on the findings. This section introduces the conclusion chapter while Section 6.1 summarizes the research done and Section 6.2 concludes this chapter.

### 6.1 The research summary

The RPL has been standardized for data transmission in Wireless devices. When the number of devices starts to grow and communicate over the internet, they form the Internet of Things (IoT) network. IoT applications comprise a large number of devices that generate a volume of data at any time. However, the RPL protocol performs poorly in large network scenarios since it lacks a load-balancing feature. The poor performance of RPL in heavy network scenarios got the attention of several researchers, who started to propose load-balancing routing algorithms for RPL protocol.

The state of the art in RPL-based load balancing routing algorithms reviewed in Chapter 3. gave clarity to the proposed RPL-load balancing routing algorithms that were perfect for balancing load distribution in different network scenarios. The algorithms' performance was evaluated in different performance metrics, which fails to clarify how these algorithms perform in another network scenario and network size. Most of the algorithms outperformed RPL in diverse performance

metrics (Kim et al. 2017; Sebastian and Sivagurunathan 2018; Taghizadeh, Bobarshad and Elbiaze 2018).

The main goal of this study is to find efficient load-balancing routing algorithms for RPL protocol in IoT networks. To achieve this goal, we proposed three RPL-based load balancing routing algorithms for IoT which utilize more than one routing metric to transmit data among IoT devices. The performance of the load-balancing routing algorithms was evaluated in the Cooja network simulator based on five performance metrics which are: Packet delivery ratio, Power consumption, End-to-End delay, Rt-Metrics, and control overhead, to find **how these load-balancing routing algorithms are effective in IoT networks**. All load-balancing routing algorithms outperformed RPL, but they all increased the control overhead. CLRPL has high overhead because its application of load balancing makes the nodes generate more control messages to notify other nodes about the congested paths to parent nodes. Therefore, CLRPL used more routes to transmit data, delayed the network, and transmitted fewer packets in the IoT network compared to lbrPL. The lbrPL transmitted more packets with high power consumption at minimum delay and routes compared to CLRPL. ETXPC-RPL underperformed in packet delivery ratio, power consumption, and delay, but mitigated the rt-metrics and control overhead more than CLRPL and lbrPL. Due to these evaluation results, it became difficult to identify the effective load-balancing routing algorithm for RPL in IoT networks. Therefore, the additional node metric, buffer occupancy was proposed to improve the performance of the load-balancing routing algorithms. The algorithms were renamed as enhanced load-balancing routing algorithms to find effective load balancing in RPL. The proposed RPL-based load-balancing routing algorithms were evaluated in the same network scenarios as the default RPL-based load-balancing routing algorithms to find the best-performing algorithm.

All the enhanced load-balancing routing algorithms outperformed the default load-balancing routing algorithms in IoT networks. Then the performance comparison of the enhanced load-balancing routing algorithms was conducted to find the best performer among all these load-balancing routing algorithms. ECLRPL outperformed BLRPL and EN-ETXPC-RPL in packet delivery ratio, became the second-best in power consumption and network delay, and the least performer in Rt-metrics and control overhead. EN- lbrPL outperformed ECLRPL and EN-ETXPC-RPL in power consumption, network delay, Rt-metrics, and control overhead and became

the second-best performer in packet delivery ratio. EN-ETXPC-RPL became the second-best in Rt-metrics and control overhead and the least performer in packet deliver ratio, power consumption, and network delay. BLRPL load balancing routing algorithm became the best performer than ECLRPL and EN-ETXPC-RPL with its ability to improve packet delivery ratio, maintain power consumption, network delay, routes used for data transmission, and control overhead. The algorithm which has a high packet delivery ratio, and less power consumption is regarded as the algorithm reliable to the network, and the algorithm which maintains the network delay and controls overhead is regarded as the stable routing algorithm to the network, therefore, the evaluation result confirms that BLRPL is the reliable RPL load balancing routing algorithm to stabilize IoT network lifetime. Therefore, the routing metrics combined for efficient load balancing routing algorithm are recommended for the future load balancing routing algorithm for IoT network scenario, Buffer occupancy, ETX, Energy-based routing metrics such as residual energy or remaining energy of the node.

## 6.2 Future Work

Routing Protocol for low power and lossy networks (RPL) is the standard routing algorithm for IoT networks, With the resource limitations of IoT devices in mind, RPL uses an objective function to route data in IoT networks. In order to find the route with the fewest bottlenecks, it uses an objective function that takes into account a wide range of parameters, including hop count, buffer occupancy, expected transmission count (ETX), bandwidth, reliability and latency. The LLNs are deployed in many different contexts, including industrial IoT, driver-less vehicles, smart cities, smart-Grid and smart-hospitals. There are applications here that require guaranteed minimum bandwidth or bounded delay. For the network to effectively transmit data for other use cases, an optimal set of metrics is necessary. Hence the effective load-balancing routing mechanism for the RPL protocol can be designed using the above-mentioned routing metrics since there is no work present in the literature to use and select the mentioned metrics dynamically in the objective function. Most of the existing load-balancing routing algorithms are performed effectively in simulation scenarios. This study also used a simulation environment to evaluate the performance of the proposed RPL-based load-balancing routing algorithm for IoT networks. We need to deploy the designed RPL-load balancing routing algorithm in a testbed. The evaluation of the outperformed enhanced load balancing routing algorithm (BLRPL) in a real scenario such as smart

farming use case where the sensors will transmit heterogeneous data traffic reading regarding the crops and environmental area can come up with a solution to close the gap in the literature. Testbed results are more reliable than the simulation result.

# Bibliography

- Abdullah, M., Alsukayti, I. and Alreshoodi, M. (2021) ‘On the Need for Efficient Load Balancing in Large-scale RPL Networks with Multi-Sink Topologies’, *IJCSNS International Journal of Computer Science and Network Security*, 21(3), p. 212. Available at: <https://doi.org/10.22937/IJCSNS.2021.21.3.29>.
- Acevedo, P.D. *et al.* (2021) ‘WRF-RPL: Weighted Random Forward RPL For High Traffic And Energy Demanding Scenarios’, *IEEE Access*, 4. Available at: <https://doi.org/10.1109/ACCESS.2021.3074436>.
- Ben Aissa, Y. *et al.* (2019) ‘QCOF: New RPL extension for QoS and congestion-aware in low power and lossy network’, *ICSOFT 2019 - Proceedings of the 14th International Conference on Software Technologies*, pp. 560–569. Available at: <https://doi.org/10.5220/0007978805600569>.
- Al-Kashoash, H.A.A., Al-Nidawi, Y. and Kemp, A.H. (2016) ‘Congestion-aware RPL for 6L0WPAN networks’, *Wireless Telecommunications Symposium*, 2016-May, pp. 1–6. Available at: <https://doi.org/10.1109/WTS.2016.7482026>.
- Al-shargabi, B. *et al.* (2020) ‘International Journal of Emerging Trends in Engineering Research Performance of RPL in Healthcare Wireless Sensor Network’, 8.
- Alayed, W., Mackenzie, L. and Pezaros, D. (2018) ‘Evaluation of RPL’s single metric objective functions’, *Proceedings - 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCoM-SmartData 2017*, 2018-Janua(June 2017), pp. 619–624. Available at: <https://doi.org/10.1109/iThings-GreenCom-CPSCoM-SmartData.2017.98>.
- Altwassi, H.S. *et al.* (2019) ‘The RPL Load Balancing in IoT Network with Burst Traffic Scenarios’, *International Conference on Software, Knowledge Information, Industrial Management and Applications, SKIMA*, 2018-Decem(February 2019). Available at: <https://doi.org/10.1109/SKIMA.2018.8631520>.
- Altwassi, H.S., Pervez, Z. and Dahal, K. (2019) ‘A burst and congestion-aware routing metric for RPL protocol in IoT network’, *2019 13th International Conference on Software, Knowledge, Information Management and Applications, SKIMA 2019* [Preprint]. Available at: <https://doi.org/10.1109/SKIMA47702.2019.8982460>.
- Arunachalam, V. and Nallamothe, N.R. (2021) ‘Load Balancing in RPL to Avoid Hotspot Problem for Improving Data Aggregation in IoT’, *International Journal of Intelligent Engineering and Systems*, 14(1), pp. 528–540. Available at: <https://doi.org/10.22266/IJIES2021.0228.49>.
- Babulak, E. (2020) ‘SL-RPL : Stability-Aware Load Balancing for RPL SL-RPL : Stability-Aware Load Balancing for RPL’, (June).
- Badach, A. (2018) ‘Routing Metrics und Constraints’, (August), pp. 1–4.
- Bagherifard, K., Ravaei, B. and Parvin, H. (2020) ‘An energy and load aware multipath routing

protocol in the Internet of Things’, (November). Available at:  
<https://doi.org/10.20944/preprints202011.0624.v1>.

Bagula, B.A. and Erasmus, Z. (2015) ‘IoT Emulation With Cooja’, *Informatics Journal*, (March), pp. 1–44. Available at:  
[http://wireless.ictp.it/school\\_2015/presentations/firstweek/ICTP-Cooja-Presentation-version0.pdf](http://wireless.ictp.it/school_2015/presentations/firstweek/ICTP-Cooja-Presentation-version0.pdf).

Basagni, S. *et al.* (2015) ‘CARP: A Channel-aware routing protocol for underwater acoustic wireless networks’, *Ad Hoc Networks*, 34, pp. 92–104. Available at:  
<https://doi.org/10.1016/j.adhoc.2014.07.014>.

Behnam Farzaneh, Ali Kamil Ahmed, E.A. (2019) ‘MC-RPL: A New Routing Approach based on Multi-Criteria RPL for the Internet of Things’, *9th International Conference on Computer and Knowledge Engineering (ICCKE)*, p. 6.

Bhandari, K.S. and Cho, G.H. (2019) ‘Resource oriented topology construction to ensure high reliability in IoT based smart city networks’, *International Journal of Systems Assurance Engineering and Management* [Preprint]. Available at: <https://doi.org/10.1007/s13198-019-00861-2>.

Bhandari, K.S., Hosen, A.S.M.S. and Cho, G.H. (2018) ‘CoAR: Congestion-aware routing protocol for low power and lossy networks for IoT applications’, *Sensors (Switzerland)*, 18(11). Available at: <https://doi.org/10.3390/s18113838>.

Bouzebiba, H. and Lehsaini, M. (2020) ‘FreeBW-RPL: A New RPL Protocol Objective Function for Internet of Multimedia Things’, *Wireless Personal Communications*, 112(2), pp. 1003–1023. Available at: <https://doi.org/10.1007/s11277-020-07088-6>.

Chen, F. *et al.* (2020) ‘Improving Topic-Based Data Exchanges among IoT Devices’, *Security and Communication Networks*, 2020. Available at: <https://doi.org/10.1155/2020/8884924>.

Clausen, T.H., Hui, J.W. and Ko, J. (2011) ‘RFC6206 : The Trickle Algorithm’, (January 2015).

Dlodlo, N., Gcaba, O. and Smith, A. (2016) ‘Internet of things technologies in smart cities’, *2016 IST-Africa Conference, IST-Africa 2016* [Preprint]. Available at:  
<https://doi.org/10.1109/ISTAFRICA.2016.7530575>.

Farag, H., Österberg, P. and Gidlund, M. (2020) ‘Congestion control and traffic differentiation for heterogeneous 6tisch networks in IIoT’, *Sensors (Switzerland)*, 20(12), pp. 1–25. Available at: <https://doi.org/10.3390/s20123508>.

Farahbakhsh, M. (2021) ‘Multi-class Multipath Routing Protocol for Low Power and Lossy Networks , with Energy Balanced Optimal Rate Assignment Multi-class Multipath Routing Protocol for Low Power and Lossy Networks , with Energy Balanced’.

Farooq, M.O. *et al.* (2015) ‘RPL-based routing protocols for multi-sink wireless sensor networks’, *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2015*, pp. 452–459. Available at:  
<https://doi.org/10.1109/WiMOB.2015.7347997>.

Farooq, M.O. *et al.* (2017) ‘Design and analysis of RPL objective functions for multi-gateway

- ad-hoc low-power and lossy networks’, *Ad Hoc Networks*, 65, pp. 78–90. Available at: <https://doi.org/10.1016/j.adhoc.2017.08.002>.
- Fatemifar, S.A. and Javidan, R. (2021a) ‘A new load balancing clustering method for the RPL protocol’, *Telecommunication Systems* [Preprint], (January). Available at: <https://doi.org/10.1007/s11235-021-00760-7>.
- Fatemifar, S.A. and Javidan, R. (2021b) ‘A new load balancing clustering method for the RPL protocol’, *Telecommunication Systems* [Preprint], (January). Available at: <https://doi.org/10.1007/s11235-021-00760-7>.
- Foubert, B. and Montavont, J. (2019) ‘Sharing is caring: A cooperation scheme for RPL network resilience and efficiency’, *Proceedings - IEEE Symposium on Computers and Communications*, 2019-June. Available at: <https://doi.org/10.1109/ISCC47284.2019.8969588>.
- Fraga-Lamas, P. *et al.* (2016) ‘A Review on Internet of Things for Defense and Public Safety’, *Sensors (Basel, Switzerland)*, 16(10), pp. 1–44. Available at: <https://doi.org/10.3390/s16101644>.
- Gaddour, O. and Koubâa, A. (2012) ‘RPL in a nutshell: A survey’, *Computer Networks*, 56(14), pp. 3163–3178. Available at: <https://doi.org/10.1016/j.comnet.2012.06.016>.
- Ghaleb, B. *et al.* (2019) ‘A New Load-Balancing Aware Objective Function for RPL’s IoT Networks’, *Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018*, (June), pp. 909–914. Available at: <https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00151>.
- Goyal, S. and Chand, T. (2018) ‘Improved Trickle Algorithm for Routing Protocol for Low Power and Lossy Networks’, *IEEE Sensors Journal*, 18(5), pp. 2178–2183. Available at: <https://doi.org/10.1109/JSEN.2017.2787584>.
- Hassani, A.E., Sahel, A. and Badri, A. (2021) ‘FTC-OF: Forwarding Traffic Consciousness Objective Function for RPL Routing Protocol’, *International Journal of Electrical and Electronic Engineering and Telecommunications*, 10(3), pp. 168–175. Available at: <https://doi.org/10.18178/ijeetc.10.3.168-175>.
- Iova, O. *et al.* (2017) ‘Using Multiparent Routing in RPL to Increase the Stability and the Lifetime of the Network To cite this version : HAL Id : hal-01206380 Using Multiparent Routing in RPL to Increase the Stability and the Lifetime of the Network \$ , \$\$’.
- Jain, J.K., Chauhan, D. and Jain, P. (2021) ‘An Energy Efficient and Bandwidth Aware Optimal Routing for IoT in Agriculture’.
- Jothikumar, C. *et al.* (2019) ‘An energy efficient cluster based routing approach to minimize energy consumption using CORP in wireless sensor networks’, *Journal of Intelligent and Fuzzy Systems*, 36(6), pp. 5835–5844. Available at: <https://doi.org/10.3233/JIFS-181658>.
- Kalyani, S. and Vydeki, D. (2018) ‘Measurement and Analysis of QoS Parameters in RPL Network’, *2018 10th International Conference on Advanced Computing, ICoAC 2018*, pp. 307–312. Available at: <https://doi.org/10.1109/ICoAC44903.2018.8939052>.
- Kiani, F. and Seyyedabbasi, A. (2018) ‘Wireless sensor network and Internet of Things in

precision agriculture’, *International Journal of Advanced Computer Science and Applications*, 9(6), pp. 99–103. Available at: <https://doi.org/10.14569/IJACSA.2018.090614>.

Kim, H. (2015) ‘QU-RPL : Queue Utilization based RPL for Load Balancing in Large Scale Industrial Applications’, *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 265–273. Available at: <https://doi.org/10.1109/SAHCN.2015.7338325>.

Kim, H.S. *et al.* (2017) ‘Load Balancing under Heavy Traffic in RPL Routing Protocol for Low Power and Lossy Networks’, *IEEE Transactions on Mobile Computing*, 16(4), pp. 964–979. Available at: <https://doi.org/10.1109/TMC.2016.2585107>.

Kim, H.S. *et al.* (2020) ‘PC-RPL: Joint Control of Routing Topology and Transmission Power in Real Low-Power and Lossy Networks’, *ACM Transactions on Sensor Networks*, 16(2). Available at: <https://doi.org/10.1145/3372026>.

Kumar, A. and Hariharan, N. (2020) ‘DCRL-RPL : Dual context-based routing and load balancing in RPL for IoT networks’. Available at: <https://doi.org/10.1049/iet-com.2020.0091>.

Latib, Z.A. *et al.* (2017) ‘A new efficient objective function for routing in Internet of Things paradigm’, *2016 IEEE Conference on Standards for Communications and Networking, CSCN 2016*, 8(4), pp. 1–6. Available at: <https://doi.org/10.1109/CSCN.2016.7785168>.

‘Literature review’ (2002) *Quality*, 41(7), p. 49.

Lodhi, M.A. *et al.* (2017) ‘Transient multipath routing protocol for low power and lossy networks’, *KSII Transactions on Internet and Information Systems*, 11(4), pp. 2002–2019. Available at: <https://doi.org/10.3837/tiis.2017.04.010>.

M.Qasem (2018) ‘Load Balancing Objective Function in RPL draft-qasem-roll-rpl-load-balancing-01’, *Standard Track*, (October), pp. 1–10.

Mamdouh, M., Elsayed, K. and Khattab, A. (2016) ‘RPL load balancing via minimum degree spanning tree’, *International Conference on Wireless and Mobile Computing, Networking and Communications* [Preprint]. Available at: <https://doi.org/10.1109/WiMOB.2016.7763185>.

Musaddiq, A., Zikria, Y. Bin and Kim, S.W. (2020) ‘Routing protocol for Low-Power and Lossy Networks for heterogeneous traffic network’.

Nesary Moghadam, M., Taheri, H. and Karrari, M. (2014) ‘Minimum cost load balanced multipath routing protocol for low power and lossy networks’, *Wireless Networks*, 20(8), pp. 2469–2479. Available at: <https://doi.org/10.1007/s11276-014-0753-7>.

Nguyen, L. and Kim, C.-K. (2019) ‘Towards Skewness and Balancing of RPL Trees for the Internet of Things’, pp. 1–12. Available at: <http://arxiv.org/abs/1903.01839>.

Onwuegbuzie, I.U., Razak, S.A. and Al-Dhaqm, A. (2022) ‘Multi-Sink Load-Balancing Mechanism for Wireless Sensor Networks’, (November), pp. 140–145. Available at: <https://doi.org/10.1109/icoco53166.2021.9673578>.

Onwuegbuzie, I.U., Razak, S.A. and Isnin, I.F. (2020) ‘Control Messages Overhead Impact on Destination Oriented Directed Acyclic Graph—A Wireless Sensor Networks Objective

- Functions Performance Comparison’, *Journal of Computational and Theoretical Nanoscience*, 17(2), pp. 1227–1235. Available at: <https://doi.org/10.1166/jctn.2020.8794>.
- Österlind, F. *et al.* (2006) ‘Cross-level sensor network simulation with COOJA’, *Proceedings - Conference on Local Computer Networks, LCN*, pp. 641–648. Available at: <https://doi.org/10.1109/LCN.2006.322172>.
- Padmaja, P.L. *et al.* (2017) ‘Role of COOJA Simulator in IoT’, *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 6(2), pp. 139–143.
- Parsaei, M.R. *et al.* (2018) ‘Rpl Load Balancing in Internet of Things’, *IJUM Engineering Journal*, 18(2), pp. 137–150. Available at: <https://doi.org/10.31436/iiumej.v18i2.702>.
- Peffer, K. *et al.* (2007) ‘A design science research methodology for information systems research’, *Journal of Management Information Systems*, 24(3), pp. 45–77. Available at: <https://doi.org/10.2753/MIS0742-1222240302>.
- Pereira, H. *et al.* (2020) ‘Increased Network Lifetime and Load Balancing Based on Network Interface Average Power Metric for RPL’, *IEEE Access*, 8, pp. 48686–48696. Available at: <https://doi.org/10.1109/ACCESS.2020.2979834>.
- Rana, P.J. *et al.* (2020) ‘Ebof: A new load balancing objective function for low-power and lossy networks’, *IEIE Transactions on Smart Processing and Computing*, 9(3), pp. 244–251. Available at: <https://doi.org/10.5573/IEIESPC.2020.9.3.244>.
- Randolph, J.J. (2009) ‘A guide to writing the dissertation literature review in software engineering’, *Practical Assessment, Research & Evaluation*, 14(13), pp. 1–13. Available at: <https://pareonline.net/pdf/v14n13.pdf>.
- Royae, Z., Mirvaziri, H. and Khatibi Bardsiri, A. (2021) ‘Designing a context-aware model for RPL load balancing of low power and lossy networks in the internet of things’, *Journal of Ambient Intelligence and Humanized Computing*, 12(2), pp. 2449–2468. Available at: <https://doi.org/10.1007/s12652-020-02382-4>.
- Sampayo, S. *et al.* (2019) ‘Selecting Parents with Wake-Up Radios for Load Balancing in RPL To cite this version : HAL Id : hal-02124314 Selecting Parents with Wake-Up Radios for Load Balancing in RPL’.
- Sankar, S. and Srinivasan, P. (2018) ‘Fuzzy logic based energy aware routing protocol for internet of things’, *International Journal of Intelligent Systems and Applications*, 10(10), pp. 11–19. Available at: <https://doi.org/10.5815/ijisa.2018.10.02>.
- Santiago, S., Kumar, A. and Arockiam, L. (2018) ‘EALBA: Energy Aware Load Balancing Algorithm for IoT Networks’, *Proceedings of the 2018 International Conference on Mechatronic Systems and Robots*, pp. 46–50. Available at: <https://doi.org/10.1145/3230876.3230880>.
- Sasaki, Y. and Yokotani, T. (2019) ‘Performance Evaluation of MQTT as a Communication Protocol for IoT and Prototyping’, *Advances in Technology Innovation*, 4(1), pp. 21–29.
- Sebastian, A. (2020) ‘Child Count Based Load Balancing in Routing Protocol for Low Power and Lossy Networks (Ch-LBRPL)’, in A.K. Somani *et al.* (eds) *Somani A., Shekhawat R., Mundra A., Srivastava S., Verma V. (eds) Smart Systems and IoT: Innovations in Computing*.

*Smart Innovation, Systems and Technologies, vol 141*. Singapore: Springer, pp. 141–157. Available at: <https://doi.org/10.1007/978-981-13-8406-6>.

Sebastian, A. and Sivagurunathan, D.S. (2018) ‘Load Balancing Optimization for Rpl Based Emergency Response Using Q-Learning’, *MATTER: International Journal of Science and Technology*, 4(2), pp. 74–92. Available at: <https://doi.org/10.20319/mijst.2018.42.7492>.

Sebastian, A and Sivagurunathan, S (2018) ‘Bandwidth Allocation based Load Balancing for RPL ( BA-LBRPL )’, (2).

Sebastian, A and Sivagurunathan, S. (2018) ‘Load Balancing Metric Based Routing Protocol for Low Power and Lossy Networks (lbrPL)’, *International Journal of Engineering & Technology*, 7(2.22), p. 39. Available at: <https://doi.org/10.14419/ijet.v7i2.22.11806>.

Sebastian, A. and Sivagurunathan, S. (2018) ‘Multi DODAGs in RPL for Reliable Smart City IoT’, *Journal of Cyber Security and Mobility*, 7(1), pp. 69–86. Available at: <https://doi.org/10.13052/jcsm2245-1439.716>.

Sebastian, A., Sivagurunathan, S. and .. (2018) ‘Load Balancing Metric Based Routing Protocol for Low Power and Lossy Networks (lbrPL)’, *International Journal of Engineering & Technology*, 7(2.22), p. 39. Available at: <https://doi.org/10.14419/ijet.v7i2.22.11806>.

Sennan, S. (2018) ‘Energy and Load Aware Routing Protocol for Internet of Things’, (October). Available at: <https://doi.org/10.11591/ijaas.v7.i3.pp255-264>.

Sennan, S. *et al.* (2020) ‘Ct-rpl: Cluster tree based routing protocol to maximize the lifetime of internet of things’, *Sensors (Switzerland)*, 20(20), pp. 1–18. Available at: <https://doi.org/10.3390/s20205858>.

Seyfollahi, A. and Ghaffari, A. (2020) ‘A lightweight load balancing and route minimizing solution for routing protocol for low-power and lossy networks’, *Computer Networks*, 179(May), p. 107368. Available at: <https://doi.org/10.1016/j.comnet.2020.107368>.

Shirbeigi, M. *et al.* (2021) ‘A Cluster-Based and Drop-aware Extension of RPL to Provide Reliability in IoT Applications’, pp. 1–7.

Shreyas, J. *et al.* (2021) ‘CAFOR: congestion avoidance using fuzzy logic to find an optimal routing path in 6LoWPAN networks’, *Journal of Reliable Intelligent Environments* [Preprint]. Available at: <https://doi.org/10.1007/s40860-021-00134-5>.

Singh, P. and Chen, Y.C. (2019) ‘RPL enhancement for a parent selection mechanism and an efficient objective function’, *IEEE Sensors Journal*, 19(21), pp. 10054–10066. Available at: <https://doi.org/10.1109/JSEN.2019.2927498>.

Sobral, J.V.V. *et al.* (2019) ‘Improving the Performance of LOADng Routing Protocol in Mobile IoT Scenarios’, *IEEE Access*, 7, pp. 107032–107046. Available at: <https://doi.org/10.1109/ACCESS.2019.2932718>.

V. Sutagundar, A., Sangulagi, P. and Tarapur, N. (2019) ‘Context Aware Load Balancing in IoT’, *International Journal of Advanced Networking Applications*, 11(02), pp. 4213–4217. Available at: <https://doi.org/10.35444/ijana.2019.11024>.

- Taghizadeh, S., Bobarshad, H. and Elbiaze, H. (2018) ‘CLRPL: Context-Aware and Load Balancing RPL for Iot Networks under Heavy and Highly Dynamic Load’, *IEEE Access*, 6, pp. 23277–23291. Available at: <https://doi.org/10.1109/ACCESS.2018.2817128>.
- Tall, H., Chalhoub, G. and Misson, M. (2015) ‘CoLBA: A Collaborative Load Balancing Algorithm to Avoid Queue Overflow in WSNs’, *Proceedings - 2015 IEEE International Conference on Data Science and Data Intensive Systems; 8th IEEE International Conference Cyber, Physical and Social Computing; 11th IEEE International Conference on Green Computing and Communications and 8th IEEE Inte*, pp. 682–687. Available at: <https://doi.org/10.1109/DSDIS.2015.107>.
- Tang, W. *et al.* (2016) ‘Toward Improved RPL: A Congestion Avoidance Multipath Routing Protocol with Time Factor for Wireless Sensor Networks’, *Journal of Sensors*, 2016. Available at: <https://doi.org/10.1155/2016/8128651>.
- Thingsquare (2014) ‘Contiki: The Open Source OS for the Internet of Things’, p. 1. Available at: <http://www.contiki-os.org/>.
- Thirupathi, V., Saritha, B. and Chitti, S. (2018) ‘Constrained Application Protocol of Internet of Things’, *International Journal of Engineering & Technology*, 7(1), pp. 244–246.
- Thomson, C. (2016) ‘Cooja Simulator Manual’, (C), pp. 2015–2016. Available at: <https://doi.org/10.13140/RG.2.1.4274.8408>.
- Varshney, K., Tripathi, S. and Purwar, V. (2021) ‘An efficient and reliable optimized routing protocol for IoT network in agriculture’, *Proceedings of the 2021 1st International Conference on Advances in Electrical, Computing, Communications and Sustainable Technologies, ICAECT 2021* [Preprint]. Available at: <https://doi.org/10.1109/ICAECT49130.2021.9392553>.
- Venable, J.R., Pries-Heje, J. and Baskerville, R. (2017) ‘Choosing a Design science research methodology’, *Proceedings of the 28th Australasian Conference on Information Systems, ACIS 2017* [Preprint].
- Winter, T. *et al.* (2012) ‘Rfc6550: Rpl’, *Internet Engineering Task Force (IETF), Request for Comments : 6550, ISSN:2070-1721*, pp. 1–157. Available at: <https://doi.org/10.1017/CBO9781107415324.004>.
- Witwit, A.J.H. and Idrees, A.K. (2020) ‘Energy-efficient Load-balanced RPL routing protocol for Internet of Things (IoTs) Networks’, *International Journal of Internet Technology and Secured Transactions*, 1(1), p. 1. Available at: <https://doi.org/10.1504/ijitst.2020.10030144>.
- Yassien, M.B. *et al.* (2021) ‘Routing protocol for low power and lossy network–load balancing time-based’, *International Journal of Machine Learning and Cybernetics* [Preprint], (0123456789). Available at: <https://doi.org/10.1007/s13042-020-01261-w>.
- Zaroor, A.R. (2020) ‘Optimizing RPL Performance Based On The Selection of Best Route Between Child And Root Node using E-MHOF method’, 11(1), pp. 25–32.
- Zhu, L., Wang, R. and Yang, H. (2017) ‘Multi-path data distribution mechanism based on RPL for energy consumption and time delay’, *Information (Switzerland)*, 8(4), pp. 1–20. Available at: <https://doi.org/10.3390/info8040124>.

