Adaptive Energy Saving and Mobility Support IPv6 Routing Protocol in Low-Power and Lossy Networks for Internet of Things and Wireless Sensor Networks

ABUBAKAR GIDADO HALILU

Submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy (PhD) in Computer Networks and Telecommunication

February 2022

Declaration

I hereby declare that no part or whole of the work referred to in this thesis has been submitted in support of an application or award of degree or qualification of this or any other university or other institute of learning. I can confirm that I take full intellectual ownership of the work submitted and the contribution of other authors or researchers within the work has been explicitly indicated with appropriate credit given. Some of the materials within this work has been published in research articles by me as the author. This work has been supervised and guided by Dr Martin Hope, and he appears as co-author on all the published articles.

List of Publications

[1] **Halilu, A. G.,** Hope, M., Takruri, H., & Aliyu, U. (2020). "*Optimized QoS Routing Protocol for Energy Scavenging Nodes in IoT*". *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 1-6. IEEE Computer Society, Porto, Portugal. July 20-22.

[2] **Halilu, A. G.,** Hope, M. (2019). *"Multipath Routing in Internet of Things using Modified Technique for Order Preference with Similarity to Ideal Solution Algorithm". The 7th Colloquium on Antennas, Wireless and Electromagnetics (CAWE),* May 2019. IET Manchester Network.

[3] **Halilu, A. G.,** Hope, M. (2019). "*Routing in Wireless Ad Hoc Networks for Next Generation Communication". Salford Postgraduate Annual Research Conference (SPARC).* University of Salford, Manchester. April 2019.

[4] Muhammad, M., Hope, M., **Halilu, A. G.,** (2018). "*Review and Analysis of a Proposed Concept for Energy Efficiency on UWB-MAC in MANET with Directional Antenna".* International Journal for Computer Applications (IJCA).

[5] Aliyu, U., Takruri, H., Hope, M., **Halilu, A. G.,** (2020). "*DS-OLSR – Disaster Scenario Optimized Link State Routing Protocol". 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 1-6. IEEE Computer Society, Porto, Portugal. July 20-22.

[6] **Halilu, A. G.,** Hope, M., (2021). *"Energy Efficiency in Low Power and Lossy Networks".* 2021 International Conference on Cyber Security Situational Awareness, Data Analytics and Assessment (CyberSA), IEEE Computer Society, South Africa June 2021.

[7] **Halilu, A. G.,** Hope, M. (2021). "*Adaptive Fuzzy Routing for 6G-Enabled Green Communication in Internet of Things". Salford Postgraduate Annual Research Conference (SPARC).* University of Salford, Manchester. June 2021.

Acknowledgement

All praises and thanks are due to Allah SWT Who in his infinite mercies has granted me the ability and strength to complete my research. Many thanks to my supervisor Dr Martin Hope, for his continued support, guidance, feedback, motivation, and exceptional positivity that helped me through the hoops and huddles of my research. Dr Hope was always keen and passionate to help in both research and professional development, and I am very proud to have him as my supervisor.

A special thanks to my wife Aisha for her unconditional love, support, and patience, she has always provided me with strength and inspiration to keep moving, and to my kids Muraad and Daneen who are my bundle of joy. Similarly, I would like to thank my family and friends for their prayers, encouragement, council, and support throughout my years of study, words cannot express my gratitude.

I would also like to thank the Petroleum Technology Development Fund (PTDF) for their valuable and precious support without which none of this would have been possible. Lastly, I would like to thank the academic staff, IT help desk, and research community at the University of Salford for their collaboration and support.

Abstract

Internet of Things (IoT) is an interconnection of physical objects that can be controlled, monitored and exchange information from remote locations over the internet while been connected to an Application Programme Interface (API) and sensors. It utilizes low-powered digital radios for communication enabling millions and billions of Low-power and Lossy Network (LLN) devices to communicate efficiently via a predetermined routing protocol. Several research gaps have identified the constraints of standardised versions of IPv6 Routing Protocol for Low Power and Lossy Networks evidently showing its lack of ability to handle the growing application needs and challenges. This research aims to handle routing from a different perspective extending from energy efficiency, to mobility aware and energy scavenging nodes thereby presenting numerous improvements that can suit various network topologies and application needs. Envisioning all the prospects and innovative services associated with the futuristic ubiquitous communication of IoT applications, we propose an adaptive Objective Function for RPL protocol known as Optimum Reliable Objective Function (OR-OF) having a fuzzy combination of five routing metrics which are chosen based on system and application requirements. It is an approach which combines the three proposed implemented Objective Functions within this thesis to enable the OR-OF adapt to different routing requirements for different IoT applications. The three proposed OFs are Energy saving Routing OF, Enhanced Mobility Support Routing OF and Optimized OF for Energy Scavenging nodes. All proposed OFs were designed, implemented, and simulated in COOJA simulator of ContikiOS, and mathematical models were developed to validate simulated results.

Performance Evaluation indicated an overall improvement as compared with the standardised versions of RPL protocols and other related research works in terms of network lifetime with an average of 40%, packet delivery ratio of 21%, energy consumption of 82% and End-to-End Delay of 92%.

Table of Contents

List of Tables

List of Figures

List of Symbols

I_{min}	Minimal interval size
$Imax$	Maximum interval size
d_f	Forward delivery ratio
d_r	Reverse delivery ratio
$\underline{D}$	Fuzzy set
$\varpi_{\underline{D}}(x)$	Membership indicator of element x in $\underline{D}$
$RSSI$	Received signal strength indicator
$Q\ (d)$	Signal loss during transmission
Z	Path loss index
X	Mean
σ	Standard deviation
d_0	Reference distance
T_P	Transmitting power
SN	Sink node
$RSSI_m$	Parent RSSI
d_{mi}	Relative distance
θ	Constant defined by hardware for LQI
β	Constant defined by hardware for LQI
N	Number of transmissions
p	Probability of success
n	Number of nodes
ψ	Probability of successful transmission
R_d	Number of successfully received packets
S_t	Total number of sent packets
T	Total duration of transmission
K	Packets transmitted
L	Packet length
R	Transmission rate
D_s	Time needed to transmit a packet
d	Distance of receiver
v	Medium characteristics

D_p	Propagation delay
D_c	Processing delay
D_{node}	Summation of $D_c D_s D_p$
D_{hop}	Hop delay
D_{e2e}	End-to-end delay
P_b	Packet loss probability
S	Number of servers
K	Packets in the system
Q	Packets in the buffer
λ	Aggregate packet arrival rate
μ	Packet departure rate
α	Traffic intensity
ρ	System utilization
π	Poisson arrival process
$E[N]$	Expected no. of packets in the system
$E[N_s]$	Expected number of packets in service
$E[D_s]$	Expected delay of packets in the server
$E[D_q]$	Expected delay of packets in the buffer
$E[D]$	Expected delay of packet in the system
$E[N_q]$	Expected number of packets in the queue
π_0	0^{th} state probability
π_i	i^{th} state probability
H_0	Initial Hop count
$E_{consumed}$	Energy consumed
I_{LPM}	Current in low power mode
I_{Tx}	Transmission current
I_{Rx}	Receive current
I_{CPU}	CPU_time current
Δt	Time between operating modes
ψ	Probability of success
B_{tt}	Battery Capacity
V	Voltage
I_o	Initial energy

RM_{energy}	Remaining energy
Sp	Step of rank
Sr	Stretch of rank
Rf	Rank factor

List of Algorithms

List of Abbreviations

AMCA	Asynchronous Multichannel Adaptation
API	Application Programme Interface
BLE	Bluetooth Low Energy
CORPL	Cognitive Routing Protocol for Low-Power and Lossy Networks
CARP	Channel Aware Routing Protocol
CoAP	Constraint Application Protocol
CAP	Contention Access Period
CFP	Contention Free Period
DIS	DODAG Information Solicitation
DIO	DODAG Information Object
DAO	DODAG Acknowledgement Object
DSME	Deterministic and Synchronous Multichannel Extension
DAG	Directed Acyclic Graph
DODAG	Destination Oriented Directed Acyclic Graph
DCCC6	Duty Cycle-Aware Congestion Control
ETX	Estimated Transmission Count
FLS	Fuzzy Logic System
FFD	Full Functioning Device
GTS	Guaranteed Time Slot
HART	Highway Addressable Remote Transducer Protocol
IEEE	Institute of Electrical and Electronics Engineering
IETF	Internet Engineering Task Force
ICMP	Internet Control Message Protocol
LRWPAN	Low-Rate Wireless Personal Area Network
LLND	Low Latency Deterministic Network
L2CAP	Logical Link Control and Adaptation Protocol
LQI	Link Quality Indicator
LLN	Low-Power and Lossy Networks
LQE	Link Quality Estimator
MCU	Microcontroller Units
M2M	Machine to Machine
MiWi	Microchip Wireless

MRHOF	Minimum Rank Hysteresis Objective Function
MTU	Maximum Transmission Unit
MQTT	Message Queue Telemetry Transport
NFC	Near Field Communication
OR-OF	Optimum Reliable Objective Function
OASIS	Organisation for the Advancement of Structured Information Standards
OMA	Open Mobile Alliance
OFDM	Orthogonal Frequency Division Multiplexing
OF	Objective Function
OCP	Objective Code Point
QoS	Quality of Service
RPL	Routing Protocol for low-Power and Lossy Networks
RFID	Radio Frequency Identifier
RAN	Radio Access Network
RFC	Request for Comments
RNFD	Root Node Failure Detection
RWP	Random Way Point
RWK	Random Walk
RSSI	Received Signal Strength Indicator
REST	Representational State Transfer
RFD	Reduced Functioning Device
ROLL	Routing Over Low-Power and Lossy Networks
SMQTT	Secure Message Queue Telemetry Transport
TSCH	Time Slotted Channel Hopping
TSMP	Time Synchronisation Mesh Protocol
TDMA	Time Division Multiplexing
UDGM	Unit Disk Graph Model
UDP	User Datagram Protocol
WSN	Wireless Sensor Networks
W3C	World Wide Web Consortium
XMPP	Extensible Messaging and Presence Protocol
Z1	Zolertia Mote
6LoWPAN	IPv6 Low-Powered Wireless Personal Area Network

Chapter One

1.1 Introduction

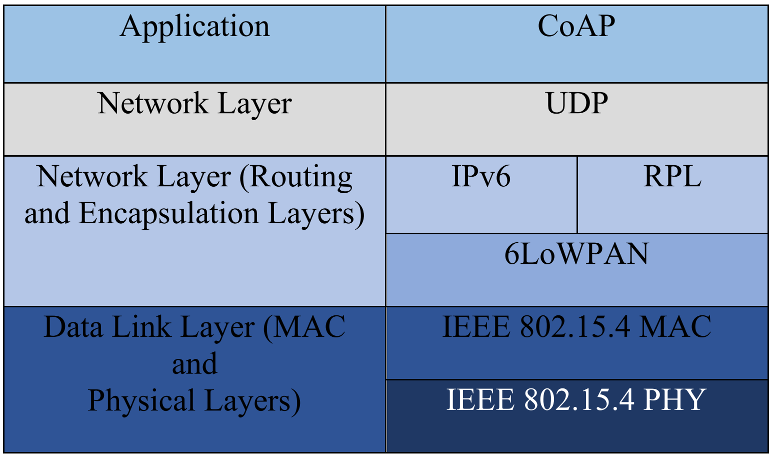
Consistent technological changes have effectively improved daily lives in many ways with the impact felt in various fields which include business, education, industrialization, healthcare, space technology and communication among others. With the internet been the most effective invention of modern times connecting millions and billions of people and devices throughout the world (Golpîra, Khan, & Safaeipour, 2021), where the evolutional shift of the internet is to network wireless objects equipped with sensors and actuators to interact seamlessly while utilizing the readily available wireless infrastructure (Gubbi, Buyya, Marusic, & Palaniswami, 2013). These objects 'things' includes a combination of various heterogenous and homogenous devices providing different kinds of services for different application needs, at the same time these devices pose a series of challenges ranging from compatibility in terms of lower layer standards to communication protocols as well as the type of device utilized.

The objects are sensors whose inherent capabilities include certain limitations in terms processing power, memory, energy, and range of transmission, with devices such as radio frequency identification (RFID) tags, Wireless Sensor devices and near field communication (NFC) devices. Internet of Things has seen huge expansion within the last decade due to ever growing innovations in both software and hardware devices as well as connections of different devices. Wireless Sensor Nodes (WSN) are often deployed in harsh or hazardous environments and utilize energy-constrained batteries for power to perform various tasks, replacing these energy sources after depletion can be a very difficult task thereby making energy saving one of the most important considerations in IoT routing. When the energy of certain nodes within the network are consumed, interruption in transmission occurs which could affect the normal working conditions of the network as well as affecting the entire network topology (Xu, Yue, & Lv, 2019).

The focus on IoT routing in network layer by most researchers is on highly establishing energy efficient paths with energy utilizing algorithms for effective forwarding mechanisms. Routing is very important in WSN as it can determine the best network path and transmission mode of data with most sensor nodes having limited energy which makes the focus on increasing network lifetime by maximizing energy utilization. Sensor nodes perform different tasks such as transmitting information related to temperature, humidity, wind speed, etc, according to their assigned roles, this information is forwarded to the base station which in turn forwards to the client system (Kamenar et al., 2016). WSNs communicate with the internet and different applications through IEEE 802.15.4 standard for low-rate wireless personal area networks (LR-WPANs) implemented on IPv6 allowing a huge number of applications to communicate seamlessly which can include smart transport, smart environment, Healthcare, smart agriculture and farming, etc. Nodes participating in IoT communication allow different applications to have varying requirements from static devices to mobile devices, from energy-constrained devices to battery operated, and much more. The potentialities of IoT are limitless as we are witnessing the ever-growing extent of IoT enabling the evaluation of routing protocols based on flexibility, reliability, and energy efficiency.

1.2 Research Motivation

IoT have witnessed an exponential growth of recent, it has seen increased improvement and development of wireless sensor devices which communicate either via an infrastructure or otherwise. IoT will allow for millions and billions of heterogenous and homogenous devices to be connected with all devices having similar characteristics of low-power, low memory, and low processing power. These challenges drive the Internet Engineering Task Force (IETF) together with other relevant bodies to come up with protocols and standards across all layers of the TCP/IP protocol stack that can conform with these application needs.



Application	CoAP	
Network Layer	UDP	
Network Layer (Routing and Encapsulation Layers)	IPv6	RPL
	6LoWPAN	
Data Link Layer (MAC and Physical Layers)	IEEE 802.15.4 MAC	
	IEEE 802.15.4 PHY	

Figure 1- 1: IoT Protocol Stack (Qasem, 2018)

Figure 1-1 depicts the IoT protocol stack with IEEE 802.15.4 MAC and PHY layers, IPv6 routing protocol for low power and lossy networks (RPL, with 6LoWPAN encapsulation), utilizing UDP communication at the transport layer and Constrained Application Protocol CoAP at the application layer (details of these protocols are explained in details in the coming chapters). RPL routing protocol is the de-facto protocol for IoT and the most widely researched routing protocol in IoT which is exponentially increasing every year (Qasem, 2018), with the number of published works increasing every year. Similarly, this research considers the exponential growth of IoT devices over the internet and tends to highlight the need for an efficient algorithm for low powered devices within the framework of existing protocols to accommodate such growth. (Lamaazi & Benamar, 2018) identified radio transceiver as one of the underlined sources of energy consumption for wireless sensors with multi-hop communication among these devices limiting the energy consumption as high-powered communication will not be needed to reach far devices. Although applications deployed in large areas employ the services of relay nodes to reach far devices such as smart environment, smart agriculture, and animal tracking.

However, RPL routing protocol is only designed for static networks without any support for mobile nodes, also having the need for various improvements in terms of energy efficiency, load balancing, support for energy scavenging nodes and mobility. Consequently, this research is driven by the inherent problems of routing in standardised RPL OF and it is aimed at enhancing the performance of RPL routing protocol by introducing three OFs and a novel adaptive OF which can suit every application need.

1.3 Problem Statement

Despite the growing interest in WSN and IoT routing protocol research for different applications having varying network needs, there is still a huge vacuum to be filled in terms of developing a suitable routing protocol that can suit the growing evolution of IoT. Default standards in IoT are best suited for static routing making the implementation of mobility routing very challenging, also most routing protocols only focus on single routing metric. Single metric routing protocols are not able to handle the overwhelmingly higher growth of IoT routing needs which creates several problems such as energy consumption, mobility management, and improved packet delivery. This research intends to handle the problems of energy consumption, mobility management, energy scavenging and delay tolerance in IoT systems.

Energy efficiency has always been an issue in sensor devices particularly those deployed in non-uniformly positioned large scale areas having unequal data traffic causing the energy of some nodes to drain faster than others. This issue mostly affects one hop neighbours of the sink node which can result in link and network failures, making energy efficiency a key design consideration in developing a suitable objective function in RPL protocol. Mobility is also another key issue in multi-hop networks, although mobility could be supported while utilizing the beacon mode in IEEE 802.15.4 standards, it is still not enough to accommodate high density networks and some specific application needs. Hence, mobility is a problem of IoT devices which will be explained further in the coming chapters.

1.4 Aim and Objectives

1.4.1 Aim

Routing is one of the major challenges in WSN and IoT applications, therefore the research will provide an adaptive RPL routing protocol to suit most application requirements that the standardised versions cannot provide. This research proposed several objective functions (OF) that provided improved performance of RPL routing protocol in terms of energy efficiency, mobility management and support for energy scavenging nodes using simulations of different antagonistic application requirements. The aim of this research can be capped in three phases; the first phase is to provide a complete simulation set of the proposed objective functions and implement them on Cooja Contiki operating system, the second phase is to ensure performance improvement in all facets as compared to the standard versions, and lastly to develop a mathematical model for the evaluation and validation of simulation results. The adaptive objective function is set to provide the growing needs of IoT applications.

1.4.2 Objective

The objectives of this research studies used to achieve the aim are as follows:

- To identify the most current research concepts, literature, and issues in RPL routing protocol while filling the research gaps.

- To provide a complete overview of our chosen simulation software for program implementation of IoT routing protocols.
- To identify the key design tools and parameters for the implementation of the proposed OF algorithm that will aim to achieve energy efficiency, handle mobility management, and provide extensive support for energy scavenging nodes.
- To provide in-depth analysis of Cooja Contiki OS which is used for the implementation aspect.
- To develop the proposed OF algorithm and evaluate with some performance metrics such as PDR, end-to-end delay, parent change, packet loss, and energy consumption with a suitable network simulator.
- To develop mathematical models that will be used for simulation validation.
- To extensively discuss and analyse results.

1.5 Research Methodology

This research adopted a quantitative scientific research as it aims to focus on simulations and mathematical models that is built on experiments, hypothesis, and proven research theories. The research methodology is displayed in Figure 1-2 and it will be a continuous process as the research progresses, the stages are outlined as follows;

- Literature Review.
- Identifying gaps/issues by carefully understanding and analysing these issues.
- Scheming of experiments and algorithms directed towards solving the problem.
- Implementation of proposed model in a simulation environment with varying node densities and scenarios.
- Developing mathematical models to validate results.
- Check solution for enhanced or improved performance
- Write PhD thesis.

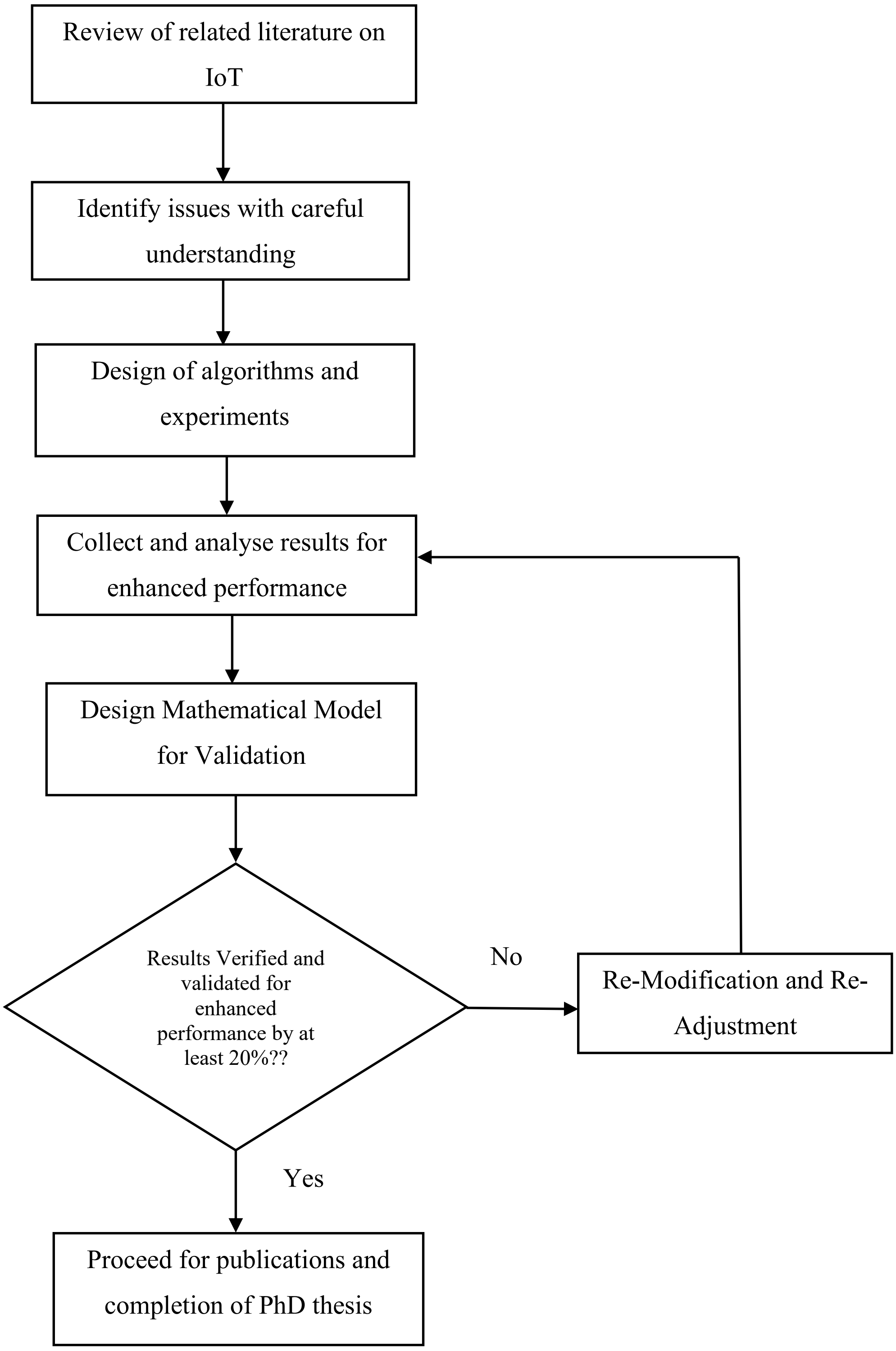


Figure 1- 2: Process Flow for Research Methodology

1.6 Research Contribution

It is believed that the importance and impact of this research will provide a better understanding of RPL routing protocol in academia as well as extending it to the Internet Engineering Task Force (IETF) standardization body. Similarly, some key contributions of this work can be summarized as follows:

- Firstly, three novel OFs were proposed with each OF having a different combination of metrics to achieve availability and better routing efficiency in terms of energy consumption, mobility management, and support for energy scavenging devices, with the aim of improving network performance.
- Secondly, an alternative approach to improve standardised RPL routing protocol without modifying the trickle timer algorithm.
- Thirdly, an adaptive routing protocol that can suit different application needs and facilitates smooth information exchange between the network and application layers, as well as optimizing the path and preferred parent selection towards the root node according to the required contextual priorities.
- Design of a single routing protocol that allows several OFs to be run simultaneously and independently of one another in the same RPL InstanceID. Also, simulations and evaluations of different performance metrics that can indicate improved performance of the proposed OFs as compared to the standardised versions of the OF.
- Lastly, implementation of individual mathematical models for validation of simulation parameters. Individual models for PDR, energy consumption, end to end delay, and mobility functions.

The thesis is structured as follows: chapter 1 highlights the nature, motivation, and objectives of the research, chapter 2 gives an overview of internet of things, chapter 3 gives related works on internet of things, chapter 4 discusses mobility support routing, chapter 5 explains enhanced energy efficiency, chapter 6 explains routing in energy scavenging nodes, chapter 7 shows implementation of fuzzy routing in internet of things and finally chapter 8 concludes the work.

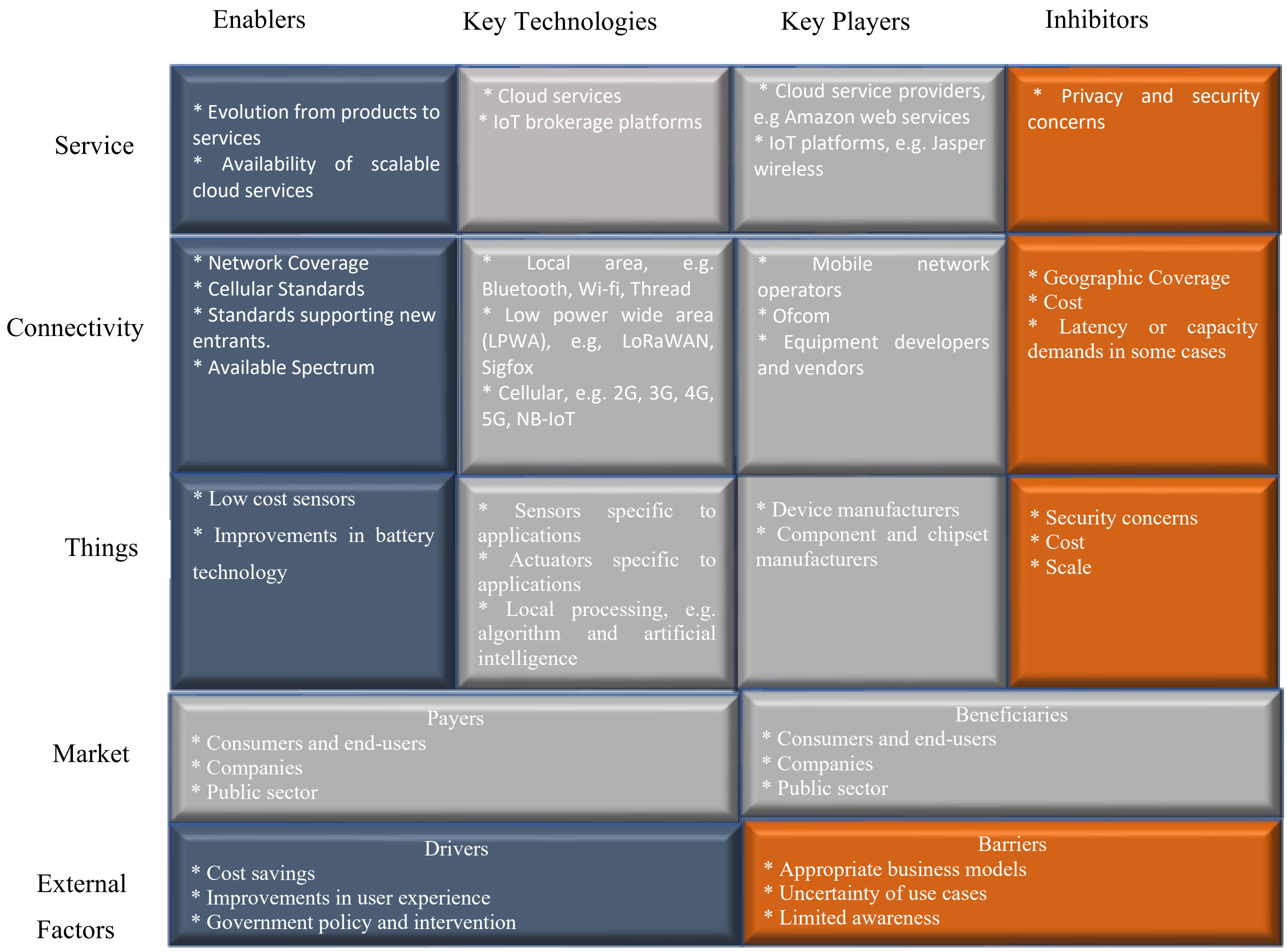
Chapter Two

Overview of Internet of Things

2.1 Introduction

Internet of Things (IoT) is an emerging area of research that is highly considered in both industry and academia. Due to the rapid evolution of smart and mobile devices, machine to machine (M2M) communication has led to the diversion of research attention to IoT enabling technologies. It is believed that IoT is presently in its prime thereby making industries invest billions of dollars in the research for IoT and its enabling technologies while also making it a research focus in academia with many expected interests in the near future(Gartner, 2014). IoT allows devices or things to connect, communicate, collaborate, and carry out decisions from remote locations. These devices might not necessarily be computers and they could either be homogenous or heterogenous. These 'Things' are miniature devices which can observe, hear, make decisions and act critically through their enabling technologies. The enabling technologies allows for ubiquitous communication and collaboration by allowing 'things' to actively make decisions, especially time critical decisions. IoT performs significantly by utilizing the underlying technologies of internet protocol, sensors, ubiquitous computing, and light communication. Although there are several challenges as well development of specialized standards and protocols for IoT, these enabling technologies must also be configured to suit IoT needs (Salman & Jain, 2015).

Figure 2-1 illustrates the various levels of the IoT ecosystem as described by (Winchcomb, Massey, & Beastall, 2017). It shows the different levels of the ecosystem providing the key enablers, technologies, players, and inhibitors at every layer of the IoT ecosystem. It gives a proper understanding of the recent trends and future directions of the IoT, including building business models by market experts to enable a suitable future investment. IoT has different operating principle on the different layers of the Transmission Control Protocol and Internet Protocol (TCP/IP) stack ranging from the application layer down to the Medium Access Control (MAC) layer. Inventions in technology has often impacted positively on modern lives in different aspects from healthcare to environment, from education to science, from military applications to outdoor activities. It is a fact that the internet provides an interconnection of numerous services that either interact with humans (H2M) or interact with themselves (M2M). Recent conception of the internet is to create services using wireless sensors and communicate seamlessly with such services and their existing technologies. IoT connections implies the integration of numerous heterogenous devices providing variety of services at the same time posing a series of challenges ranging from standardization, to differential application requirements as well as limitation in terms of the capacity of devices (Stankovic, 2014). Wireless Sensor Networks (WSN) consist of



	Enablers	Key Technologies	Key Players	Inhibitors
Service	* Evolution from products to services * Availability of scalable cloud services	* Cloud services * IoT brokerage platforms	* Cloud service providers, e.g Amazon web services * IoT platforms, e.g. Jasper wireless	* Privacy and security concerns
Connectivity	* Network Coverage * Cellular Standards * Standards supporting new entrants. * Available Spectrum	* Local area, e.g. Bluetooth, Wi-fi, Thread * Low power wide area (LPWA), e.g, LoRaWAN, Sigfox * Cellular, e.g. 2G, 3G, 4G, 5G, NB-IoT	* Mobile network operators * Ofcom * Equipment developers and vendors	* Geographic Coverage * Cost * Latency or capacity demands in some cases
Things	* Low cost sensors * Improvements in battery technology	* Sensors specific to applications * Actuators specific to applications * Local processing, e.g. algorithm and artificial intelligence	* Device manufacturers * Component and chipset manufacturers	* Security concerns * Cost * Scale
Market	Payers * Consumers and end-users * Companies * Public sector		Beneficiaries * Consumers and end-users * Companies * Public sector	
External Factors	Drivers * Cost savings * Improvements in user experience * Government policy and intervention		Barriers * Appropriate business models * Uncertainty of use cases * Limited awareness	

Figure 2- 1: IoT Ecosystem (Winchcomb et al., 2017)

devices with certain limitations in terms memory, processing power, coverage distance, energy, and physical size. WSNs provide enabling standards of internet protocol network communication for Low-Rate Wireless Personal Area Network (PAN) with IEEE 802.15.4 (ZigBee, Wireless HART, MiWi, 6LoWPAN, Thread, ISA100.11a and SNAP) (Khalil, Abid, Benhaddou, & Gerndt, 2014). These standards allow for miniature devices to actively participate in internet communication. Some applications of WSNs include emergency operations, transportation, industry, battlefield, healthcare, smart homes, agriculture, etc.

IoT has the capacity to accommodate different applications with varying requirements from real time applications to data sensitive applications. However, designing an effective and efficient routing algorithm involves consideration for robustness, minimum delay, enhanced energy efficiency, scalability, and adaptability. IoT has shown tremendous potentials in improving lifestyle with its relative ease in communication and interaction. There are increasing number of advents in the research of WSNs as well as reduction in the cost and maintenance of these devices. Energy efficiency in sensor networking is one of the key considerations in designing IoT. By 2025 there will be significant increase in IoT and may be applicable to everyday things such as paper documents, food packages and much more, however, all these achievements will pose future risks and security concerns when deceives are illegally accessed, monitored and located from remote locations (Villaverde et al., 2013). Similarly, increase in demand and future technological advances will see IoT contributing immensely to modern military warfare and economic developments. Interconnecting these 'things' with the internet will produce a huge connection uniquely addressing communicating devices. The internet has had its revolution on mankind and still making a huge impact in so many aspects but combining such revolution with the concept of IoT paves way for further mind-blowing revolutions particularly with information processing, communication, resource sharing, information aggregation and provision of services (Yang, Yang, & Plotnick, 2013).

2.2 IoT Applications

The proliferation of IoT growth has seen a huge potential in developing different applications in different fields of research. The progression of IoT research and implementation in areas such as Smart Homes, transportation, Healthcare, smart environment, Military, etc, has been very satisfying despite some challenges and potential complexities that can be envisioned (Al-Fuqaha, Guizani, Mohammadi, Aledhari, & Ayyash, 2015). Additionally, IoT has the potential for disaster/emergency management using real time applications. Real-time applications make use of the time-driven concept of IoT, where events are monitored continuously over a certain interval of time. Contrary to the event-driven concept that is triggered in the event of an emergency or sudden occurrence. According to (Winchcomb et al., 2017), IoT applications need to satisfy the following criteria,

- Endpoints must be embedded in everyday objects
- Endpoints must use an embedded microprocessor
- Endpoints must connect via the internet
- Endpoints must use interconnected networks
- Endpoints must use standardised communications.

The above points define what an IoT is or rather what requirements are needed for an IoT device to satisfy.

2.3 IoT Forecast Model

According to a recent study conducted by Arne Holst from Statista in 2021, that by 2030, China will have approximately 7.7 billion IoT devices, with several other regions such as Europe and North America are expected to have a great number of IoT connections as displayed in Figure 2-2;

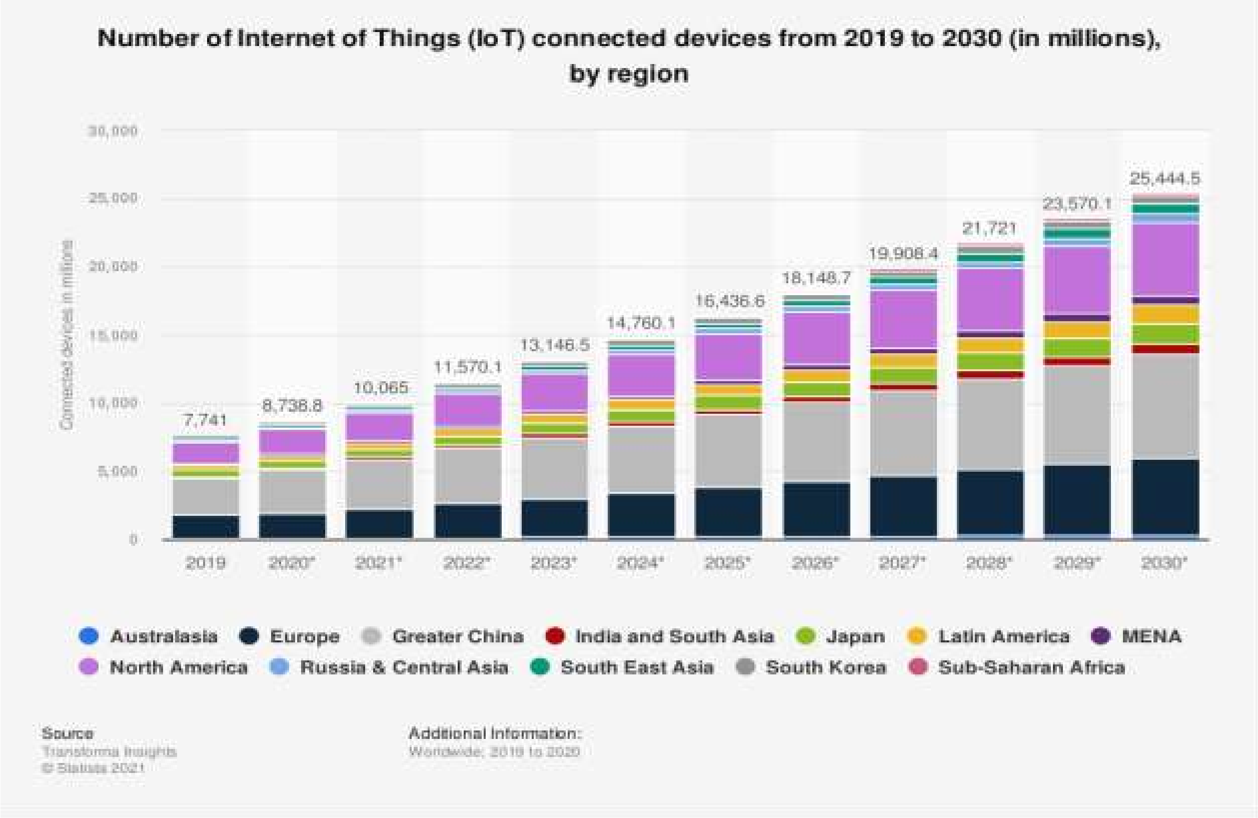


Figure 2- 2: An overview of the forecast model between the end of year 2019 and end of year 2030

2.4 Protocols for IoT

The organisation of the IoT protocol stack is slightly different compared to the traditional Open System Interconnect (OSI) or the Transmission Control Protocol/Internet Protocol (TCP/IP). In the OSI reference model, there are seven layers, namely; Application, Presentation, Session, Transport, Network, Data Link, and Physical. Additionally, with the TCP/IP model, three top layers are merged into one layer taking the number of layers down to five.

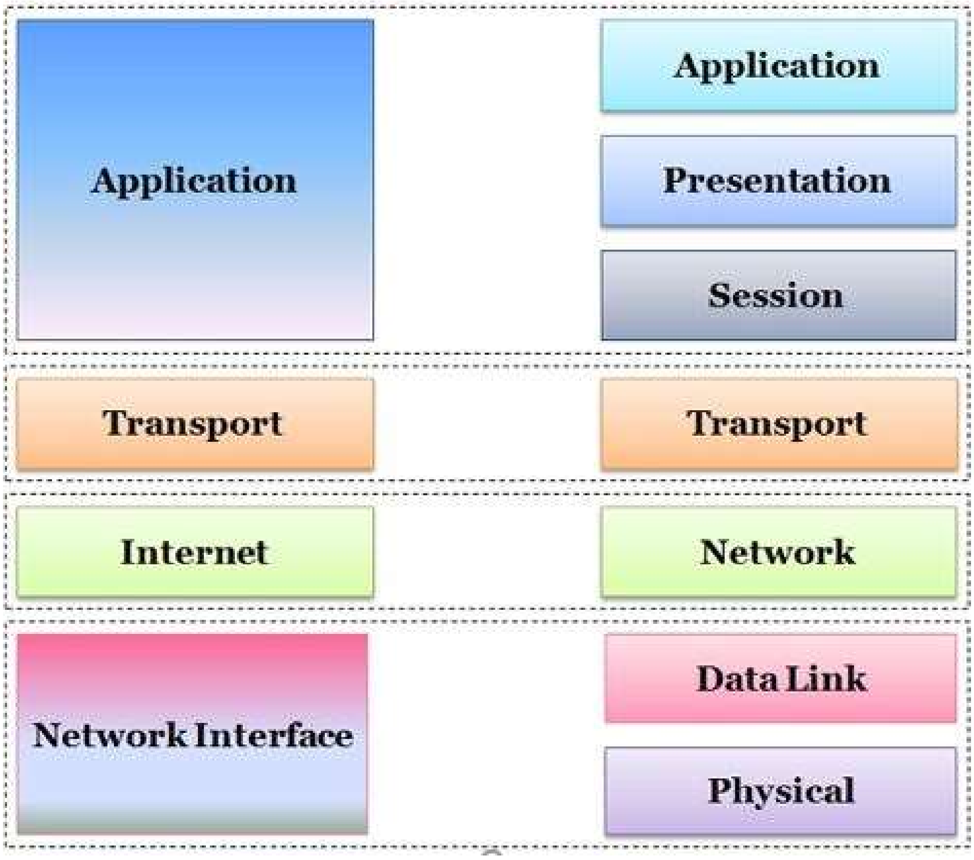


Figure 2- 3: OSI and TCP/IP Protocol Stack (Y. Li, Li, Cui, & Zhang, 2011)

However, for IoT to fit in properly into the protocol stack, new standards needed to be designed to accommodate constrained devices by standardization bodies such as Internet Engineering Task Force (IETF), Organisation for the Advancement of Structured Information Standards (OASIS), Open Mobile Alliance (OMA) and World Wide Web Consortium (W3C) (Poudel, 2016). A breakdown of the varying differences between the traditional protocol stacks and IoT stack is given in Figure 2-3.

In the above models, the OSI model has seven complete layers with each layer having a different functionality in terms of performance and operations with the previous layer. Similarly, in the TCP/IP model there are five layers of interconnection where the session and presentation layers are merged into application layer. However, the structure of the IoT layers has been miniaturized as can be seen in the Figure 2-4. Figure 2-4 (Qasem, 2018), shows the various layers in the IoT protocol from the session layer to the data link layer. These protocols allow for low end and resource constraint devices to access the internet and consequently enabling life-altering solutions. The main standard protocol for IoT which was adopted from WSN is the IEEE 802.15.4 that specifies the acceptable communication and interoperability framework of low powered devices defining the Physical and Data Link layers of the protocol. The protocols can be discussed further in the coming sections.

Session	**MQTT, SMQTT, CoRE, DDS, AMQP, XMPP, CoAP, …**		**Security**	**Management**
Network	Encapsulation	6LowPAN, 6TiSCH, 6Lo, Thread, …	TCG, Oath 2.0, SMACK, SASL, ISASecure, Ace, DTLS, Dice, …	IEEE 1905, IEEE 1451, …
	Routing	RPL, CORPL, CARP, …		
Data Link	Wi-Fi, Bluetooth Low Energy, Z-Wave, ZigBee Smart, DECT/ULE, 3G/LTE, NFC, Weightless, HomePlug GP, 802.11ah, 802.15.4e, G.9959, WirelessHART, DASH7, ANT+, LTE-A, LoRaWAN, IEEE 802.15.4 MAC, IEEE 802.15.4 PHY, …			

Figure 2- 4: Detailed IoT protocol Stack

2.4.1 IEEE 802.15.4

The IEEE 802.15.4 defines standards and specifications for both the physical and data link layer of the models of low end and energy constraint devices. It serves as the foundation by which higher layers of the protocol stack build upon and allows for suitable frame formats for low end multi-hop devices to be transmitted. To achieve higher reliability while simultaneously meeting communication requirements for IoT, it utilizes time synchronization and channel hopping. Some of the features of IEEE 802.15.4 can be summarized below (Salman & Jain, 2015);

Slot frame Structure: This frame structure used by IEEE 802.15.4e gives instructions to each node aside from its scheduling responsibilities. It tells nodes when to perform the basic operations of sleep, send and receive. In the sleep mode, nodes naturally save power by turning off their radios, while in transmission it sends data and awaits acknowledgement from the recipient, and finally when receiving, the radio is turned on for the data to be received, forwarded to the upper layers, and acknowledged before it's turned back off.

Scheduling: This MAC feature specifies how scheduling is achieved to accommodate different mobility scenarios. The manager node develops the schedule and furnish member nodes with instructions on how the schedule will be followed.

Synchronization: This feature enables nodes to maintain real time connectivity with their neighbouring nodes and the gateway. In acknowledgement-based synchronisation, receipt of acknowledgement serves as a guarantee for connection reliability in communicating nodes, while in frame-based synchronisation, empty frames are sent at pre-defined times to ensure synchronisation.

Channel Hopping: This is a MAC feature in IEEE 802.15.4 that allows channels to be changed at certain time intervals using a pre-defined random sequence to allow for smooth access to the wireless medium. The aim is to minimize interference by selecting from sixteen available channels and to reduce multi-path fading by forwarding at least two frames on different channels over the same network link.

Network Formation: This feature specifies the method of advertising and joining the network. For potential devices trying to join the network, they must first listen to the advertisement before sending their joining request. Network formation is deactivated once a device has fully joined the network and it is only re-activated upon receiving another joining request from intending nodes. Requests received from centralized systems are processed by a manager node, while request from decentralized systems are processed locally.

The application requirements for IEEE 802.15.4 allows for network devices to operate as either a Full-Function Device (FFD) or Reduced-Function Device (RFD) (H. D. Y. Kharrufa, 2018). The FFD has the full functioning capability of all network operations and can be a Personal Area Network (PAN) coordinator, local coordinator, or regular node, while RFD due to its low resource's constraint has limited functionalities handling low level applications. FFD and RFD can use different available network topologies for IEEE 802.15.4, the network topologies are flat, chain, cluster tree topologies (Velmani & Kaarthick, 2014). The selection of the PAN coordinator by member nodes is based on the device with FFD capabilities, and different applications perform differently in certain network topologies.

Flat Topology allows nodes to communicate directly to one another, it involves continuous exchange of packets from one node to the other by a process known as flooding. This topology does not save energy despite having a very simplified concept and causes a lot of overhead (Velmani & Kaarthick, 2014). The flat network topologies as reviewed in the literature are (Heinzelman, Kulik, & Balakrishnan, 1999; Intanagonwiwat, Govindan, & Estrin, 2000; Yao & Gehrke, 2002). In Chain Topology, some nodes are selected as gateways through which other nodes can communicate. This is in effort to reduce flooding the network by member nodes but causing other issues like communication delays by bottom chain nodes willing to communicate with higher level nodes (Lindsey & Raghavendra, 2002; Shin & Suh, 2011). Cluster-based topology is flexible and scalable with energy saving capabilities (Goyal & Tripathy, 2012). In this type of network topology, a leader is selected from among the nodes known as a Cluster Head (CH), the selection usually puts into consideration the suitability of a node for instance, its distance to base station, number of neighbours, remaining energy, transmission range, etc. Nodes from different clusters communicate with each other through the CH (Bandyopadhyay & Coyle, 2003; Heinzelman, Chandrakasan, & Balakrishnan, 2000; Younis & Fahmy, 2004).

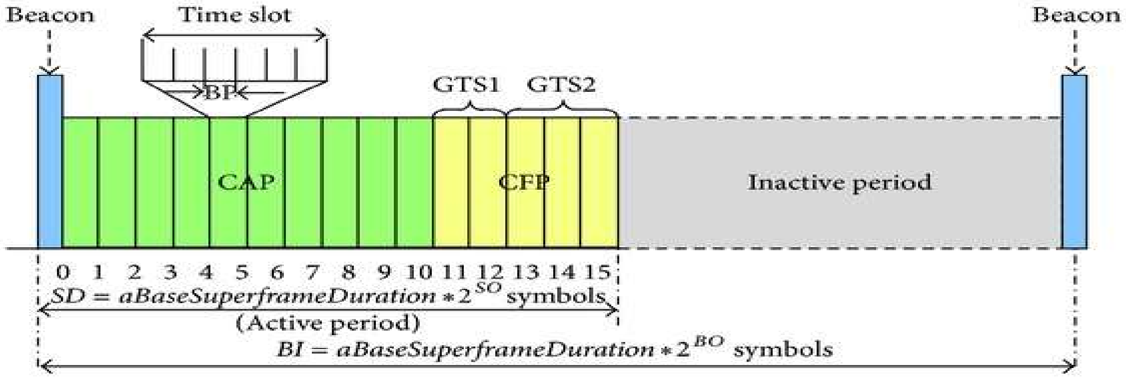


Figure 2- 5: Super Frame Structure (Xia, Li, Hao, Kong, & Gao, 2013)

The physical layer of IEEE 802.15.4 is responsible for ensuring that proper communication is established and maintained, it ensures channel detection/selection and radio transmission/reception. Whereas the MAC layer ensures channel access and node association/dissociation (H. D. Y. Kharrufa, 2018). It ensures beaconing where an elected PAN coordinator maintains connectivity by forwarding periodic beacons to connected nodes. The first time slot for transmission of the super frame is mostly reserved for the beacons, while the remaining time slots are for Contention Access Period (CAP) and Contention-Free Period (CFP). From Figure 2-5, devices communicate with slotted CSMA/CA in CAP utilizing the super frame time slots, while using CFP to avoid end-to-end delays of CSMA/CA utilizing the Guarantee Time Slots (GTS), devices also utilizes the inactive periods to enter sleep mode thereby saving energy. The IEEE 802.11 designed for wireless access has a light low energy version designed specifically to incorporate IoT technology known as IEEE 802.11ah. Due to the overhead in frame transmission and power consumptions in the former, it is not suitable for use in IoT applications, while the latter supports very low communication overhead and low power consumption, it is highly suitable for IoT sensors (Park, 2015). Another data link protocol that utilizes IEEE 802.15.4 PHY in the physical layer and Time Division Multiple Access (TDMA) on the MAC layer known as Wireless Highway Addressable Remote Transducer Protocol (Wireless HART). It offers both reliability and integrity by encrypting messages moving across wireless devices, routers and adapters (A. N. Kim, Hekland, Petersen, & Doyle, 2008; Raza & Voigt, 2010). Z-Wave is another MAC protocol specifically designed for IoT applications in home automation (smart home and small commercial domains). It utilizes point-to-point communication for IoT applications like heartbeat monitor, light control, temperature control, etc (Sinha & Park, 2017). Bluetooth Low Energy (BLE) is mostly utilized for in-vehicle networking and short distance communication. It has very low latency and fast data transmission utilizing both advertising and data frames for communication between member nodes and master nodes respectively (Gomez, Oller, & Paradells, 2012).

ZigBee smart energy is like Z-Wave in terms of purpose in home automation, remote control systems and E-Health. It is also designed to accommodate low energy IoT applications while supporting different types of available network topologies in IoT. It uses two modes of operation which is normal ZigBee and ZigBee Pro, both modes support full low powered networking while ZigBee Pro offers better security, scalability and better performance in multicast routing (Salman & Jain, 2015). DASH7 is also another MAC wireless protocol used by IoT applications. It has better performance than normal ZigBee in terms of range, scalability and data rate (Cetinkaya & Akan, 2015). Another communication standard defined for IoT applications in cellular networks and Machine to Machine communication is Long-Term Evolution Advanced (LTE-A). It is a low-cost protocol using OFDMA and suitable for IoT communications. It consists of Core Network (CN), Radio Access Network (RAN) and Mobile Nodes. CN controls devices and RAN establishes wireless connectivity. Communication between CN and RAN is via S1 link (Hasan, Hossain, & Niyato, 2013), Other MAC protocols for IoT include HomePlug GreenPHY (Alliance, 2012), G.9959 (ITU), LoRaWAN (Thubert, Pelov, & Krishnan, 2017), Weightless (Poole, 2014), Digital Enhanced Codeless Communication/Ultra Low Energy (DECT/ULE) (Bush, 2015).

2.5 Network Layer Protocols

The network layer protocols have been divided into two for easier identification and explanation. The first layer is the routing layer which is responsible for forwarding packets from source to destination, while the second layer is the encapsulation layer which is responsible for building up the packet. Most of earlier versions of routing protocols developed specifically for the WSN and MANET using mobile nodes thereby utilizing heavy amount of data exchange without necessarily considering the energy consumption of nodes. These also contributed to the idea of developing low power devices by the Routing Over Low power and Lossy networks (ROLL) working group, composing of devices that are cheaper and constrained in nature (in terms of processing power, memory, energy) with a multipoint to single point traffic pattern. LLNs also have the inherent ability to be unstable due to connectivity losses, with low throughput from connected nodes. (Kushalnagar & Montenegro, 2007) explained that a lot of routing protocols existed prior to the standardization of LLNs and most of these protocols were not properly suited for LLNs due the following reasons;

- Resource Constraint/Awareness (consideration of limited resources in routing such as the remaining energy, memory, and link quality).
- Utilization of Multipath (selection of multiple parents by the forwarding mechanism).
- Consideration of Maximum Transmission Unit (MTU) (MTU plays a significant role in LLN but the limitation of the MTU were mostly disregarded by the previous protocols, thus limiting the MTU by LLN in MAC layer increases the chances of receiving a frame successfully by transmitting shorter frames.)

- Density awareness (previous protocols utilize flooding in discovering and maintaining routes, but this practice is limited in LLNs to avoid total network failures and broadcast storms)
- Tiny footprints (LLN implementation of RAM and ROM are in few kilobytes and tens of kilobytes respectively, having unsuitable capabilities for effectively storing routes and neighbour tables)
- Low Power Mode (LLN manages power by intermittently switching the radio on and off as the situation requires, where existing protocols mostly assumes devices are on 'ON' mode)

Some defined LLNs routing protocols were divided into reactive and proactive protocols. The reactive protocols were built based on the adherent nature of the Ad Hoc on Demand Distance Vector routing protocol (AODV) of the traditional WSN known as 6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD) (Thubert & Hui, 2011). Although the development of the protocol was suited for the LLN in that it optimizes performance while reducing the size and frequency of the control message transmission during the route discovery and route maintenance processes. It is loop free in that relay nodes need not send reply, only destination nodes respond with a reply message. This protocol was adopted by first edition ITU-T G.9903, while (Clausen, Yi, & Herberg, 2017) adopted an improved version utilizing custom metrics for the third edition ITU-T G.9903 known as Lightweight On-Demand Ad Hoc Distance Vector Routing Protocol for Next Generation (LOADng). The Inner workings of the LOAD and LOADng can be summarized as thus;

- When a router initiates transmission to a destination, it sends a *Route Request* (RREQ) message as a broadcast message to all its neighbours which is like the concept of traditional AODV. Upon receiving a RREQ packet by a neighbour node, it further relays the RREQ to its neighbours thereby attaching relevant information such as RREQ ID, reverse route, sequence number etc, to ensure the same packet is not received more than once by any intermediate node, with the aim of preventing network breakdown. This way, any intermediate node that receives the same packet automatically drops the packet.
- Once a RREQ is received by the destination it has two options, either to send a *Route Reply* (RREP) immediately or wait for designated time for other RREQ from

neighbour nodes (i.e. when destination node has more than one neighbour), so that it chooses the reply route with the minimum cost metric.

- Similarly, when destination node sends RREP through the intermediate nodes, each traversed node will acknowledge and construct a forward route as the packet is transmitted to the source. Acknowledging receipt of packets by intermediate nodes prevents unidirectional links from been included in the forwarding route while ensuring that links are active.
- *Route Error (*RERR) is only originated if a next hop is unreachable by a node after its failed attempt to repair the route locally. Once it is received by the sender, a rebroadcast is initiated.

The proactive protocol develop by the Internet Engineering Task Force (IETF) is the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) (Vučinić, Tourancheau, & Duda, 2013). It is built on the concept of Gradient-based routing which entails building a network with specific consideration to the height of a node using either their hop-count (HC) or Expected-Transmission Counts (ETX) to forward traffic to the sink. The RPL routing protocol is the backbone of this research and it will be discussed in more details in the coming sections. Other IoT routing protocols include Cognitive RPL (CORPL) which is developed for cognitive LLN devices and an extended version of RPL having few modifications, and Channel-Aware Routing Protocol (CARP) for underwater communication.

2.5.1 RPL

The IPv6 Routing Protocol for Low Power and Lossy Networks has become the de factor and most widely researched routing protocol for IoT. It allows LLNs and its emerging technologies to conveniently exchange information by providing the much needed enabling infrastructural support from the lower layers of the protocol stack as depicted in Table 2-6. The adoption of the protocol by Zigbee Alliance as identified by (Iova, Picco, Istomin, & Kiraly, 2016) together with IEEE 802.15.4 MAC and Physical Layers indicates a satisfactory level of acceptance. As stated by (Qasem, 2018), the ROLL used the concept of (Winter, 2012) to develop the RPL standardization (Popa, Salazar, Dejean, & Jetcheva, 2011). RPL was developed as a distance vector proactive routing protocol or table-driven protocol, and proactive routing protocols naturally ensures continuous updates of routing information in both storing and non-storing modes thereby avoiding loops and stalled routes. Its Link layer robustness allows it to work with varying technologies as well as transmitting in different traffic patterns P2P, P2MP and MP2P. The typical organisational structure of RPL routing involves a central node which can be the root node (or a boarder router) utilizing the low-bandwidth and fluctuating links to establish communication which is like the case of Smart Meters, Advanced Metering Infrastructure or Power Line Communication for Narrowband. The standardised network architectures by the ROLL working group are Building, Home, Urban and Industrial Networks, with various recommendations for standard behaviours according to the ROLL template of operational applicability.

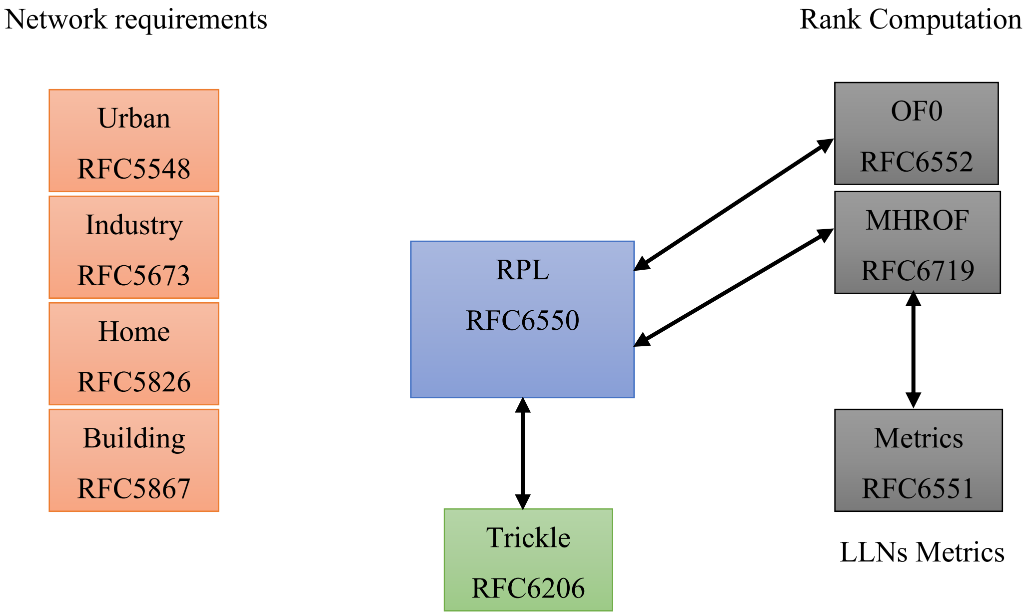


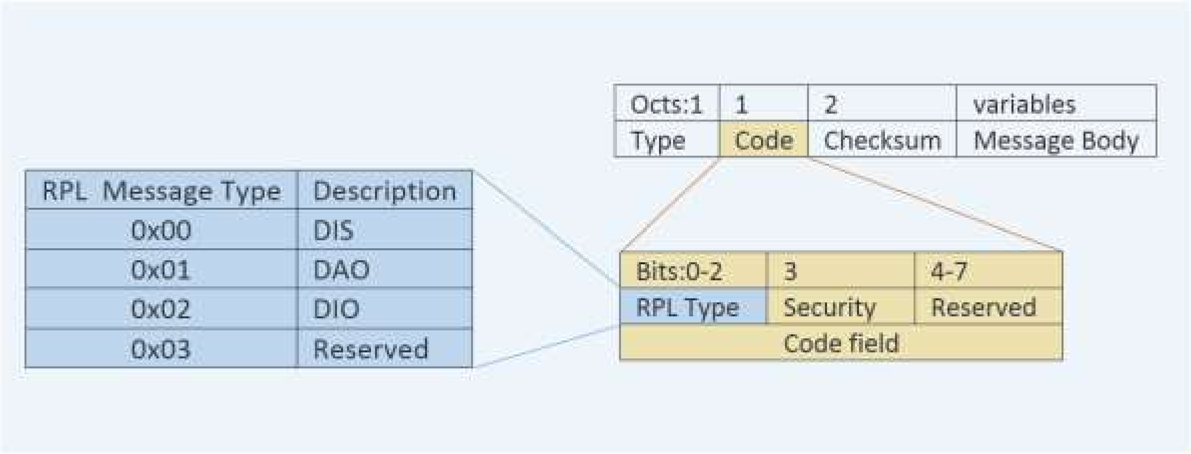
Figure 2- 6: RPL Ecosystem (Dujovne, Watteyne, Vilajosana, & Thubert, 2014)

These templates also include constraints particularly with network heterogeneity and the type of application been utilized. (Popa et al., 2011) specified the draft requirements for Advanced Metering Infrastructure which was proposed to be standardised to an RFC. RPL supports three security modes Unsecured, preinstalled and authenticated, the first mode means that RPL does not provide any additional method of security and been unsecured does not necessary mean that the network is unsecured, it might be using link-layer protection. The second method ensures that nodes joining an RPL Instance must have an existing preinstalled key for generation and processing of messages, and lastly the authenticated mode where nodes with preinstalled key can join the RPL Instance as child nodes while the authenticated authority provides the required key for joining as a router.

2.5.1.1 Construction of RPL Topology

The fundamental nature of LLNs makes it difficult for them to maintain a specific predefined topology, this is due to their innate ability to always change their route selection process so that the constructions of routes from source to destination is continuous. The construction of the route discovery phase starts with the creation of a Directed Acyclic Graph (DAG). A DAG is a finite graph defined in set theory as both directed and acyclic, it contains vertices and edges such that no vertex or edge will be looped. Each DAG contains at least one root node with no outgoing edges. During route construction, RPL utilizes the Destination Oriented DAG (DODAG) to build routes to the destination (root), implying that a DODAG is a special DAG where each member node wants to reach a single destination. A collection of one or more DODAG which signifies a group of directly connected nodes sharing the same Objective Function (OF) is identified by an RPLInstanceID. Similarly, each DODAG is uniquely identified by a DODAGID within an RPLInstance with each new shape of DODAG representing a new DODAG version, and each RPLInstance is identified by its RPLInstanceID.

A goal is where each DODAG wants to reach, when a DODAG reaches its destination it is said to be grounded whereas when a DODAG is not connected or is yet to reach its goal its known as floating. An Objective Function on the other hand is something that was proposed to be minimized that helps to decide the distance to the root and it is mostly decided by the programmer using a metric or a combination of metrics.



Octs:1	1	2	variables
Type	Code	Checksum	Message Body

RPL Message Type	Description
0x00	DIS
0x01	DAO
0x02	DIO
0x03	Reserved

Bits:0-2	3	4-7
RPL Type	Security	Reserved
Code field		

Figure 2- 7: RPL ICMP Message Format (Gaddour & Koubâa, 2012)

Communication in RPL involves an exchange of a series of control messages in discovering and maintaining of routes, these control messages are DODAG Information Object (DIO), DODAG Information Solicitation (DIS), DODAG Advertisement Object (DAO), DAO-ACK and consistency checks that deals with security. The control messages as represented by (Gaddour & Koubâa, 2012) can be seen in Figure 2-7;

The RPL message types can be briefly summarized as thus (Qasem, 2018; Ropitault, Pelov, Toutain, Vedantham, & Itron, 2015):

- **DIO:** This message is multi-casted downwards or by any given node in a DODAG who wishes to inform other nodes about its presence. The information transmitted within the DIO may include things like whether the node is storing or non-storing, whether its grounded or not, or its connectivity status with other nodes in the DODAG. These configurable parameters used within an RPL instance is classified under the following, an RPLInstanceID, objective function in use, rank, DODAG parent set, DODAGID and DODAG root identity. Normally, the parent selection metrics and DODAG construction information are sent periodically with a sequence number for the selection process.

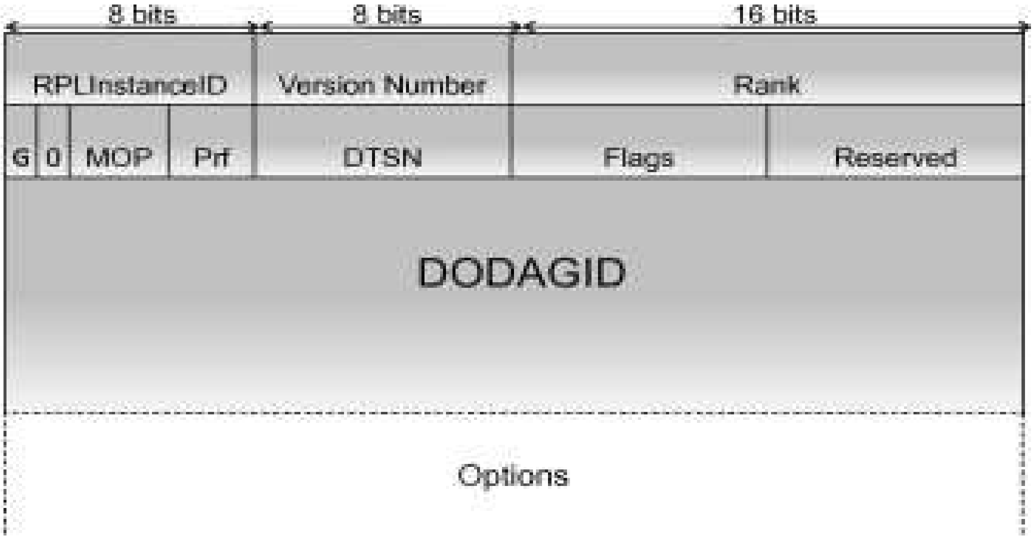


Figure 2- 8: DIO Message Format (Gaddour & Koubâa, 2012)

- **DIS:** This control message is sent by nodes who are interested in joining a DODAG or when a node belonging to a DODAG has not heard an announcement after a period, so it sends the message to know if any DODAG exists.
- **DAO:** This is sometimes considered as an optional feature for P2MP and P2P applications, but when enabled, it indicates a response from a DIO transmitted message sent in form of a DAO message indicating the destination information towards the upward DODAG.

  It is a request sent by a child node to a selected parent node in storing mode requesting to join the DODAG, while in non-storing mode messages are sent directly to the DODAG root which is always in storing mode. DAO messages are also sent by RPL to indicate node prefix for preferred parent in downward route formation.
- **DAO-ACK:** This is the DODAG Advertisement Acknowledgment sent in response to the DODAG Advertisement Object message. It is an optional feature which can either be enabled or disabled to allow forwarding unicast message to the sender of the DAO message expressing their disposition to allow messages pass through them as a next hop or as a DODAG root.

The DIO message is often used to create a DAG by the DODAG root node (LBR- Low Power and Lossy Network Border Routers) and it encompasses relevant configuration parameters with settings that is used for path and preferred parent discovery in an RPL Instance as well as maintaining the DODAG. From Figure 2-8, the RPL Instance ID displays the identification of the DODAG to which the Instance belongs to, Version number is a continuously incremented number of a DODAG that aids in ensuring updated information is propagated across nodes in the network, Rank indicates the rank of the node with respect to the DODAG root node, DTSN (Destination Advertisement Trigger Sequence Number) specifically used for downward routes, Grounded (G) is a flag which shows the connectivity status of the DODAG, MOP (Mode of Operation) specifies the type of operation adopted by the RPL Instance from among the four available operations (storing, non-storing, support for downward routes and multicast), Preference (Prf) specifies preferential levels for DODAG root and DODAGID is the IPv6 unique address for DODAG root (Gaddour & Koubâa, 2012).

2.5.1.2 Route Formation and Rank Construction in RPL

There are two ways by which routes are formed in RPL, the upward and the downward route formations, and these two route formation principles follows either of the three known communication patterns. The normal communication from DODAG root to member nodes is considered as P2MP, communication from member nodes to DODAG root is considered as MP2P and communication within a DODAG considered as P2P. The two categories of route formation will be discussed below;

Formation of Upward Routes: When an RPL network is formed, one of the nodes in the network will be the DODAG root where all information is directed toward, this node can either be the Sink node or a boarder router. The DODAG root initiates the broadcast of the DIO message to its directly connected nodes containing its rank and other relevant information, the gradient-based feature of RPL allows it to obey the monotonic principle in forwarding information in the network. When a DIO message is received by a node, it uses the DIO Rank field to compute its rank together with the chosen OF. While the OF determines which selected metric to minimize, the outcome is used in rank calculation enabling the node to include its route in the DIO sender to the RPL root. The process of route formation utilizing the DIO mechanism from members of a DODAG to DODAG root is known as upward route formation, and each DODAG member node is said to have a minimum of one upward route for it to belong to that DODAG. The most suited communication pattern for upward route formation is the MP2P where all communication is directed to the DODAG root after the rank computation by member nodes.

Figure 2-9 depicts a typical upward route formation layout. The root node initiates the DIO message and send it to its immediate neighbours (i.e. nodes within its radio transmission range), these nodes receive the DIO message and process it thereby adding the root node as their preferred parent. Similarly, neighbour nodes forward the processed DIO messages to nodes within their radio transmission range and the same processing is repeated until the complete route is formed.

Formation of Downward Routes: Downward routes are formed from the DOAG root to nodes in an RPLInstance. Construction of downward routes are optional and can be achieved in either storing or non-storing modes using DAO messages. When in storing mode, the preferred parent inserts the downward routes upon reception of the DAO message, where the processed DAO message is continually forwarded until it reaches the root node. Conversely in non-storing mode, DAO messages are forwarded straight to the root node without the preferred parent inserting the downward routes. The intermediate nodes just serve as relay nodes and source routing is used by the root node to forward information in this mode. While still utilizing the hop-count as the selected metric in a storing mode, the preferred parent inserts the downward route upon receiving a DAO message from a node. This process continues until the DAO is finally received by the root node.

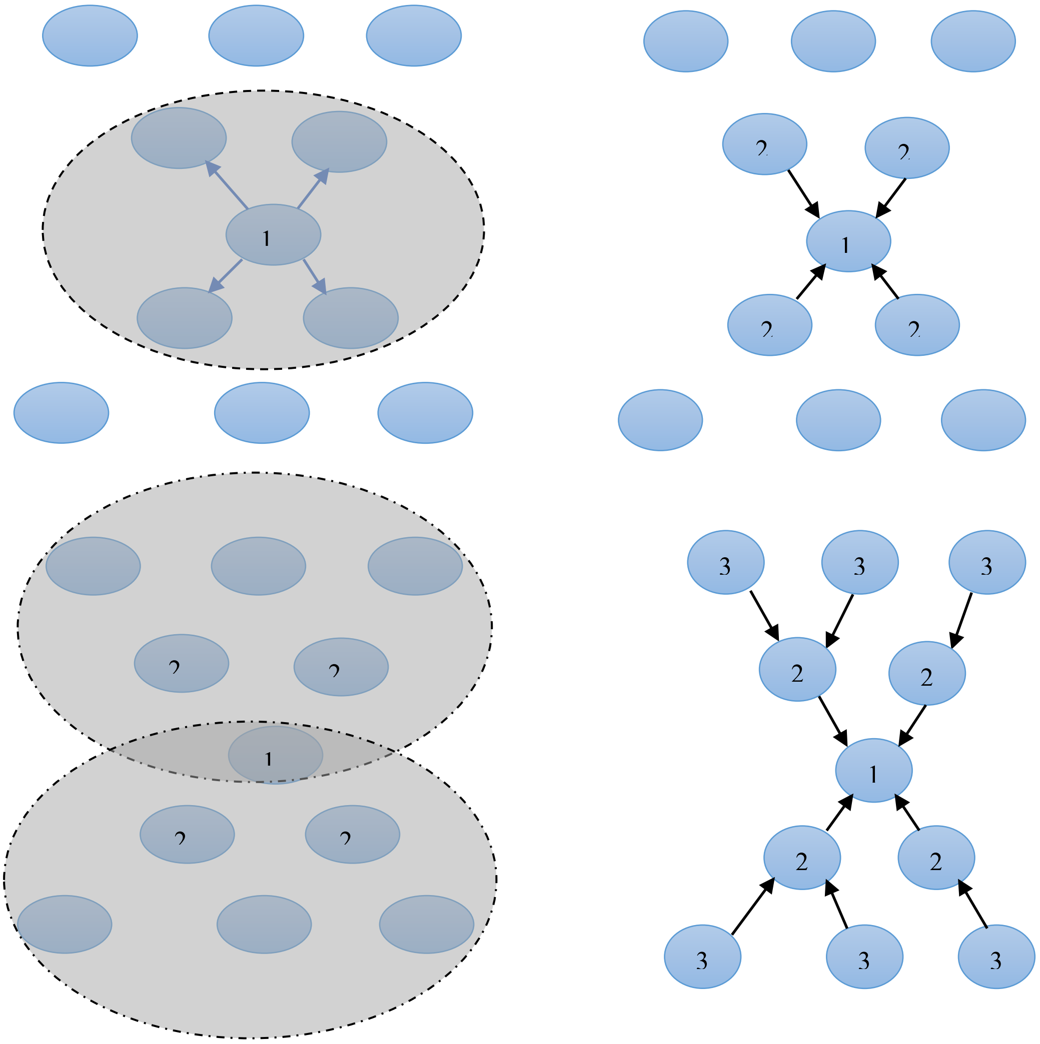


Figure 2- 9: Route Formation in RPL (Ropitault et al., 2015) (the grey circle represents the transmission range of the node; the small circles indicate the node and the numbers indicate the node rank with respect to the sink node)

Whenever a network outage occurs or network failures, there is need to classify the problem into complete failures (global) or simply failures associated with a certain section. For complete failures, repairs are initiated by the root node which seems a lot like forming a new DODAG altogether. Except that in this case, only the DODAG version number is incremented, and nodes are expected to transmit their locations. In contrast to the local repair, it does not affect the whole DODAG but only the selection of a preferred parent and nodes are able to select their preferred parent irrespective of their ranks.

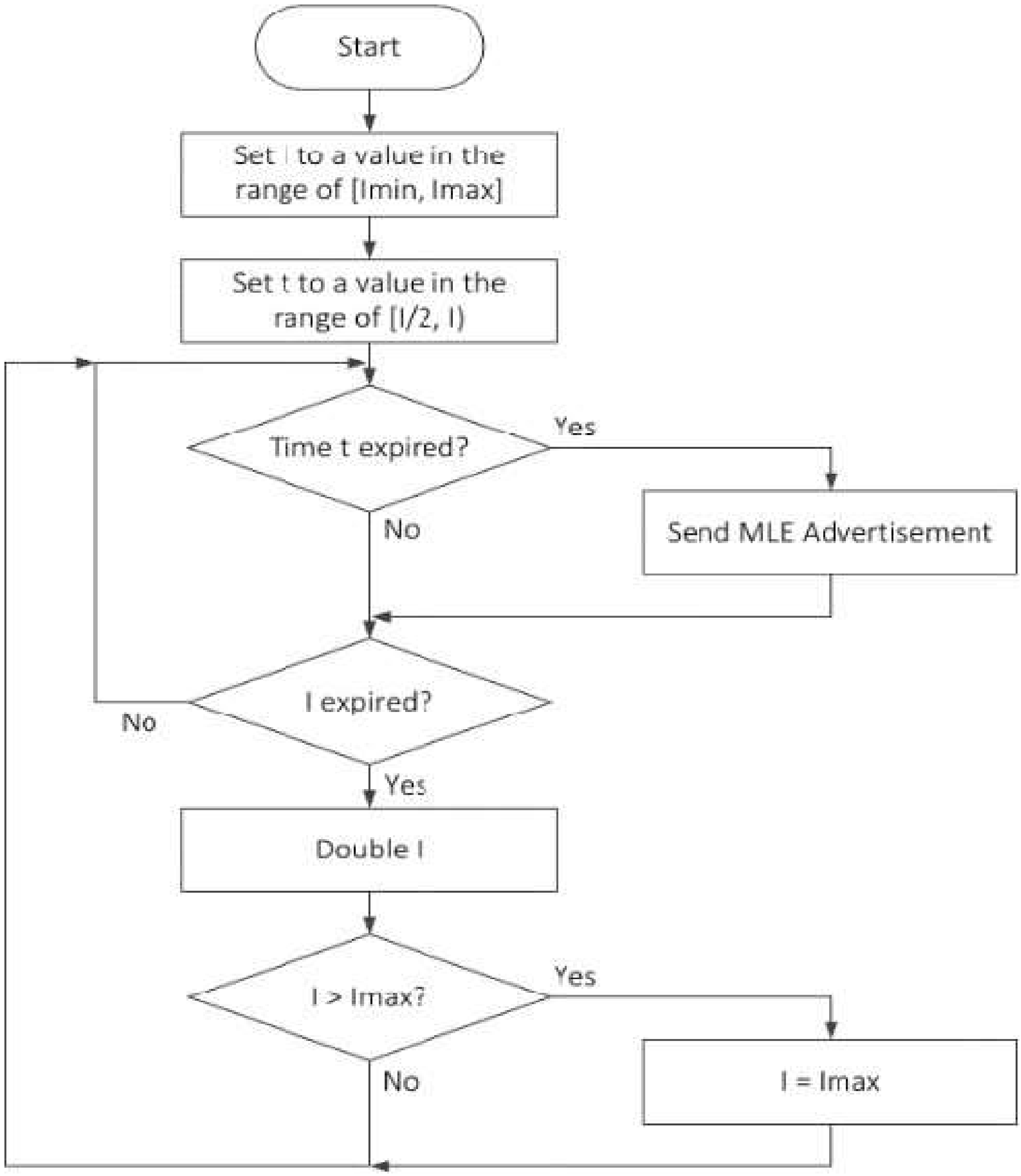


Figure 2- 10: Process Flow for Trickle Timer Algorithm (Sikora, Schappacher, Zimmermann, & Mars, 2017)

2.5.1.3 Trickle Algorithm

Low-power and lossy networks in a shared medium utilize the trickle algorithm to communicate efficiently by exchanging information in a highly robust, scalable and energy efficient manner. The trickle algorithm continuously adjusts transmission window to ensure updated information are always circulated to the nodes, and its communication rate tends to increase logarithmically with respect to the nodes density using meek suppression mechanism with point selection (Caicedo-Ortiz et al., 2018). Using the density-aware mechanism coupled with a consistency model it provides a transmission guide to nodes, for instance, when node's data are consistent with one another, the control messages are sent infrequently with just very few packets per hour, but when node's data are inconsistent with its neighbours, the trickle algorithm quickly sends control messages in milliseconds to resolve the issue. It controls the frequency of control message exchange in the network and automatically adjusts the network parameters during disconnection, loss, or network re-population. Due its extreme flexibility, it can handle a wide range of related issues like control of multicast propagation, quick propagation of updates, new route discoveries, even distribution of transmission load, imposing low maintenance overhead, ability to handle robust and slight network disconnections, handling different variations of network density and control of traffic timing.

Using trickle algorithm, nodes transmits data upon overhearing other transmissions in the network that makes its transmission seem redundant, for instance, software update versions, last received packet or other routing state information. Analysing a trickle message by nodes can be of two types, either a node ascertain that the data is consistent with its own or inconsistent, for example, if a node *C* transmits its version number as *VN*, but *D* already has version number of *VN + 1*, automatically *D* recognises that *C* needs an update. Also, if *D* transmits *VN+1* version number to *C, C* realises that it requires an update, and *D* transmits *VN+1* to its immediate neighbours without the nodes advertising their need whereby any inconsistencies will be detected and updated (Caicedo-Ortiz et al., 2018). Trickle timer algorithm transmission frequency is determined by the level of inconsistencies in the network routing information, it is higher in case of inconsistencies and reduced in situations of stable networks.

The trickle algorithm can be properly described using the flowchart in Figure 2-10.

As defined by (Levis & Clausen, 2011), the algorithm can be explained as thus:

- The trickle algorithms run using 3 defined intervals, the minimum interval size *Imin* which specifies time in milliseconds or seconds, the maximum interval size *Imax* which is defined by the base-2 log(max/min) representing the required doublings of minimum interval size, and *k* redundancy constant which is an integer or a natural number.

- The variables *I, t* and *c* for current interval size, time within current interval and counter respectively are set.
- Upon the start of the algorithm, it sets *I* within the range of *Imin* and *Imax*. Sets *C* to *0, t* from a range *[I/2, I]* from a random point.
- When trickle realises that the transmission is consistent, the counter is incremented. While trickle transmit only if the redundancy *k* is greater than the counter *c* at any given time *t*. *K* is needed to suppress DIO transmission.
- Upon the expiration of *I* the interval length is doubled by the trickle and tested with the *Imax*, if it is longer than the time for *Imax*, trickle sets *I* to be the time specified by *Imax*. (MLE- maximum likelihood estimation is used by sink nodes to advertise changes in routing table to immediate neighbours)
- When trickle perceives an inconsistency in transmission with *I* greater than *Imin*, it resets the trickle timer with *I* as *Imin*.

2.5.1.4 Objective Function

The Objective Function (OF) is one of the most important features in RPL by enabling it to compute a node's rank which is the distance to the root node as well as defining how routes are selected and optimized in an RPLInstance. The root node encapsulates within its DIO message an Objective Code Point (OCP) in other for nodes to know the OF being used within the RPLInstance.

<table>
<tr><td colspan="10">0</td><td colspan="10">1</td><td colspan="10">2</td><td colspan="2">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Type</td><td colspan="8">Opt Length</td><td colspan="4">Flags</td><td colspan="1">A</td><td colspan="3">PCS</td><td colspan="8">DIOIntDoubl.</td></tr>
<tr><td colspan="8">DIOIntMin.</td><td colspan="8">DIORedun.</td><td colspan="16">MaxRankIncrease</td></tr>
<tr><td colspan="16">MinHopRankIncrease</td><td colspan="16">OCP</td></tr>
<tr><td colspan="8">Reserved</td><td colspan="8">Def. Lifetime</td><td colspan="16">Lifetime Unit</td></tr>
</table>

Figure 2- 11: DODAG Configuration Format (Winter, 2012)

The OCP is embedded in the DODAG configuration which is used to transmit configuration information within a DODAG. Figure 2-11 represents the DODAG configuration file format and the two OF defined by ROLL are Objective Function Zero (OF0) and Minimum Rank with Hysteresis Objective Function (MRHOF).

Figure 2-11 is used to transmit configuration information concerning the nature of the adopted DODAG operation of an RPL Instance. While the remaining information can be static, this research ensured that the *OCP* remains dynamic for every implementation of an OF or a change in the adopted routing OF. The information is only modified by the DODAG root node and the fields in the configuration file can be explained as thus: Type indicates the nature of message as defined by IANA to mean a number of routing information labelled from *0x00 – 0x09 (Pad1, PadN*, *DAG* metric container, Routing Information, DODAG configuration, RPL Target, Transit Information, Solicited Information, Prefix Information and Target Descriptors) (Winter, 2012). Length indicates the length of the selected octets information, flags mostly remaining unused by the sender and can be initialised or otherwise, '*A'* indicates Authentication Enabled used for security is the network indicating whether authentication is required when a node wants to join the network, *PCS* (path control field) is used to specify the path control field and *DIOIntervalDoublings* used to calibrate the *Imax* of trickle timer DIO.

DIOIntervalMin is used to calibrate the *Imin* of trickle timer *DIO, DIORedundancyConstant* used to calibrate the constant value of the trickle timer algorithm, *MaxRankIncrease* used to calibrate the increase in rank that will be used in local repair, *MinHopRankIncrease* used to configure the minimum rank increase, *OCP* specifies the OF used, *Reserved* kept for future use, *Default Lifetime* used for RPL paths given as in seconds as (Def. Lifetime x Lifetime Unit) and *Lifetime Unit* indicates RPL route lifetime which could either be in hours or sometimes days (Winter, 2012).

2.5.1.4.1 Objective Function Zero (OF0)

OF0 as defined in (Thubert, 2012) uses hop-count as its deterministic function and it works simply by utilizing the addition of a range of scalar values (from 1 excellent to 9 very poor) to compute the rank of the preferred parent (as specified in the DIO message). To determine the rank of a node, it is computed by adding a preferred parent rank to the rank increase whereby the rank increase is an expression of a link property which can be shown mathematically as thus (Ropitault et al., 2015);

$$R\ (N)\ =\ R\ (P)\ +\ rank\_increase \qquad \mathbf{2.1}$$

$$rank\_increase\ =\ (Rf\ \times\ S_p\ +\ S_r)\ \times\ MinHopRankIncrease$$

where *R(P)* is the rank of the preferred parent, *Sp* is *step_of_rank* which is the expression of link property between 1 and 9, *Sr* is *stretch_of_rank* which is the maximum extension to the *Sp* of a potential parent needed to allow possible selection of successor. It is device-specific, if none is configured then *Sp* is not stretched. *Rf* is *rank_factor* which is used to increase the usefulness of the link properties. The *MinHopRankIncrease* is an important feature in rank computation as it shows how a given metric affects rank increase, a bigger value indicates high impact. In addition to *MinHopRankIncrease* been used in the parent selection, its multiplicative nature impacts the number of hops in the RPLInstance. There are different rules in parent selection, but it is always important to select that which has a lesser resulting rank.

2.5.1.4.2 Minimum Rank Hysteresis Objective Function (MRHOF)

MRHOF is an objective function as defined by (Gnawali, 2012) which tends to optimize paths that minimizes a metric at the same time avoiding paths that changes metrics frequently with just small variations by introducing hysteresis (path history). DIO metric container allows DIO messages to be transmitted in an additive manner by the MRHOF as messages move along routes in an RPLInstance. MRHOF uses path cost which is a quantified value of a route to the RPL root with regards to the metric used. It obtains the path cost by adding the link metric cost to the preferred parent and the path advertised by the parent. The goal of an OF is to convert metric into rank. Rank computation can be expressed in equation 2.2 and 2.3:

$$path_{cost}\ =\ parent_{path\_cost}\ +\ link\_cost \qquad \mathbf{2.2}$$

$$rank\ =\ func\ (\ path_{cost}) \qquad \mathbf{2.3}$$

where *link_cost* is the cost associated with the parent's link, $parent_{path\_cost}$ is transmitted by the parent, while turning a *path_cost* to rank depends on the chosen metric. The *PARENT_SWITCH_THRESHOLD* is used as the hysteresis function which is the minimum difference between the path cost to the preferred parent and the potential parent to initiate the current preferred parent selection. The ETX selects routes with high throughput, and it can be expressed in equation 2.4:

$$ETX = \frac{1}{d_f \times d_r} \qquad \mathbf{2.4}$$

Where *dr* specifies the reverse delivery ratio, and *df* specifies forward delivery ratio. While the probability that a transmission is received and acknowledged is $d_f \times d_r$. The idea of objective function is to support RPL in fulfilling different optimisation requirements in dealing with LLN applications.

There are several categories of routing metrics used in IoT applications as defined by (Barthel, Pister, Dejean, Vasseur, & Kim, 2012) which can be used for path computation, the routing metric is considered as a quantitative value that can be used to determine the path cost. The selection of the best path is most suited for paths that satisfy all the given conditions or constraints and usually the one that has the least cost with regards to some specified metrics. Most routing metrics for LLNs are additive in nature and they can be branded using the following characteristics, qualitative versus quantitative, dynamic versus static and link versus node metrics.

Table 2- 1: IoT OS and its Implementation (Ropitault et al., 2015)

S/N	***Operating Systems***	***Mode***	***Objective Function***	***Metric Type***
1.	***TinyOS RPL***	*Storing*	*OF0 and MRHOF*	*Hop-count and ETX*
2.	***ContikiRPL***	*Storing*	*OF0 and MRHOF*	*Hop-count and ETX*
3.	***SimpleRPL (Linux)***	*Storing*	*OF0*	*Hop-count*
4.	***RIOT***	*Non-storing and Storing*	*OF0 and MRHOF*	*Hop-count and ETX*
5.	***OpenWSN***	*Non-storing and Storing*	*OF0 and MRHOF*	*Hop-count and ETX*
6.	***Unstrung (Linux)***	*Non-storing and Storing*	*OF0 and MRHOF*	*Hop-count and ETX*

The most frequently used routing metric is the link versus node metric, the link related metrics are: throughput, latency, link quality, expected transmission count and link colour, while the node related metrics are node state and attribute, node energy and hop-count. The different operating systems in IoT having various implementation of metrics can be summarized in Table 2-2 (Ropitault et al., 2015).

Different operating systems exist for the implementation of IoT applications and different simulation environments are also available such as Qualnet, WSNet, Castalia, OMNET++, NS3/NS2, JSIM, COOJA, etc., all these simulation environments offer the implementation codes for running IoT applications.

Several researchers use these simulators/emulators to represent variety of scenarios depending on their business or application needs. (Chauvenet, Tourancheau, Genon-Catalot, Goudet, & Pouillot, 2010) used a Power line communication (PLC) in developing a smart metering architecture utilizing the 6LoWPAN in IEEE 802.15.4 MAC, and the application is tested using COOJA simulator. Similarly, (Saad, Chauvenet, & Tourancheau, 2011) utilized COOJA simulator to analyse the impact of root node mobility in WSN with specific regard to the effects of energy consumption as compared to a static node both participating in PLC. The proposed Signal-to-noise Interference-Ratio (SINR) by (Shelby, Hartke, & Bormann, 2014) was also implemented in ContikiOS with COOJA simulator, the model focused on mainly physical layer characteristics implemented on IEEE P1901.2 in using intermediate voltage power lines.

2.6 IoT Application Layer Protocols

2.6.1 Message Queue Telemetry Transport

Message Queue Telemetry Transport (MQTT) is an internet transfer protocol which was standardised by OASIS (Karagiannis, Chatzimisios, Vazquez-Gallego, & Alonso-Zarate, 2015). It utilizes the publish/subscribe network architecture by ensuring connectivity between applications/middleware’s and networks. Within the IoT context, the publishers of the messages are those lightweight sensors that are already connected to a broker, which is an intermediary between the publishers and the subscribers. Subscribers on the other hand refer to those applications that require the services of those sensors, and they get this information from the broker. Some of the tasks performed by the brokers include the classification of different sensory data into topics which are sent to interested subscribers (Salman & Jain, 2015).

2.6.2 Secure Message Queue Telemetry Transport

Secure MQTT is an encryption-based extension of MQTT for lightweight IoT applications. It ensures the broadcast of encrypted messages to multiple nodes using the broadcast encryption feature. During the set-up phase, each publisher and subscriber are expected to declare their interested area to the broker, who uses a key generation algorithm to develop the secret key to match their areas of interest. Thereafter, the encrypted data is published by the broker to interested subscribers, who use their generated secret key to decrypt the message. The SMQTT is designed solely to improve the security aspects of MQTT, however both encryption and key generation algorithms has not been standardised (M. Singh, Rajan, Shivraj, & Balamuralidhar, 2015).

2.6.3 Extensible Messaging and Presence Protocol

Extensible Messaging and Presence Protocol (XMPP) was standardised by IETF since 2009 and it has been recently utilized in software defined networks for IoT applications. Its originally intended purpose was for chatting and messaging, it is very efficient over the internet and uses extensible mark-up language (XML) standards. It supports low-latency messages with a near real time application supporting two network architectures which are Publish/subscribe and request/response that can be determined by the network administrator. However, additional overhead from XML message headers are generated thereby increasing the amount of processing required by the application at the same time affecting the power consumption making it unsuitable for QoS applications and highly impractical for machine to machine communication (Saint-Andre, 2011).

2.6.4 Constrained Application Protocol

Constrained application protocol (COAP) is an internet transfer protocol designed specifically for resource constrained applications by the IETF working group. This is one of the standards or protocols for messaging in IoT, and there are several other messaging functions in IoT utilizing either TCP or UDP for communication. The different functions are interoperable, and they are often referred to as the session layer protocols. COAP is designed by Constrained RESTful Environment (CORE) in other to provide a lightweight RESTful HTTP interface (Salman & Jain, 2015). It enables low-powered devices to utilize RESTful services for communication, Representational State Transfer (REST) between client and server providing interface for resource constrained devices to interact. The architecture of COAP is either for messaging or request/response for message reliability and communication respectively.

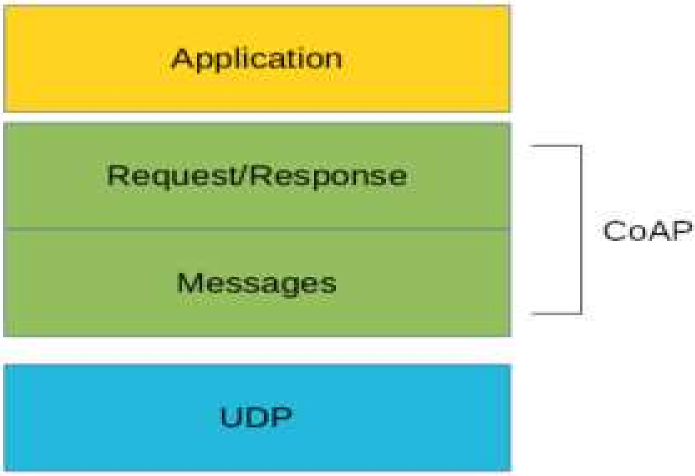


Figure 2- 12: Layers of Constrained Application Protocol (R. Chen, Guo, & Bao, 2014)

The modes of COAP messaging as specified by (Salman & Jain, 2015) is divided into: piggyback, confirmable, non-confirmable and separate. TCP and UDP transmissions represent the confirmable and non-confirmable respectively. Piggyback is a request/response message between client and server, where the client sends message directly to the server and the server acknowledges. While in separate mode, the server takes some time in responding to a message sent by the client and such response may come in a different message from the acknowledgement (Shelby et al., 2014).

Table 2- 2: Comparing IoT Application Layer Protocols (Salman & Jain, 2015)

S/N	***Protocols***	***Connection***	***Architecture***	***QoS and Security***	***Header Size (Bytes)***
1.	***DDS***	*TCP/UDP*	*Publish/Subscribe*	*QoS*	-
2.	***CoAP***	*UDP*	*Request/Response*	*Both*	*4*
3.	***AMQP***	*TCP*	*Publish/Subscribe*	*Both*	*8*
4.	***MQTT***	*TCP*	*Publish/Subscribe*	*Both*	*2*
5.	***XMPP***	*TCP*	*Both*	*Security*	-

From the message models of COAP presented earlier, each message contains a unique ID during transmission. Although some messages may arrive differently but including message markings on the headers ensures that transmitted messages must only be acknowledged if the same message ID is received. Therefore, retransmission only occurs during a timeout or when the retransmission counter has not reached the maximum transmission limit. Other protocols include Advanced Message Queuing Protocol (AMQP) (Standard, 2012) and Data Distribution Service (DDS) (Rodríguez-Molina, Bilbao, Martínez, Frasheri, & Cürüklü, 2017).

2.7 IoT Network Layer Encapsulation Protocols

2.7.1 6LoWPAN

This is the most common standard for IoT applications, it stands for IPv6 over Low power Wireless Personal Area Network. It supports low bandwidth and different IoT networking topologies, encapsulates, or reduces the long headers of IPv6 to IoT IEEE 802.15.4 standards, reduces power consumption and supports high mobility. The reduced packet headers do not exceed 128 bytes achieved through fragmentation of frames which in turn minimizes the transmission overhead with full multi-hop delivery support. The four types of headers used in 6LoWPAN are (Salman & Jain, 2015); No 6LoWPAN header with the number 00 where frames are discarded for failure to meet the 6LoWPAN requirements, Dispatch header with the number 01 specifically used for multicasting, Mesh header with the number 10 used exclusively for broadcasting, and finally the Fragmentation headers with the number 11 used in breaking long headers into128 bytes and arranged into fragments. 6LoWPAN serves as a bridge residing between network and data link layers continuously transforming IPv6 to IEEE 802.15.4 standards for communication. Although IEEE 802.15.4 standard has an overhead of 25 bytes from the available 128 bytes leaving 103 bytes of payload with much further reduction if security measures are put in place. Additionally, the use of UDP in IPv6 further reduces the payload header with 48 bytes bringing the number of remaining usable payload to an even lower number (H. D. Y. Kharrufa, 2018).

Figure 2-13 shows the levels of communication and connectivity between the TCP/IP protocol networks and the IEEE 802.15.4 based networks. RFC 6282 defines security architectures that could allow a usable payload header of up to 108 bytes (Thubert & Hui, 2011), with RFC 4944 performing arrangements of fragmented headers exceeding the maximum transmission unit (MTU) specified by IEEE 802.15.4 standards (Kushalnagar & Montenegro, 2007).

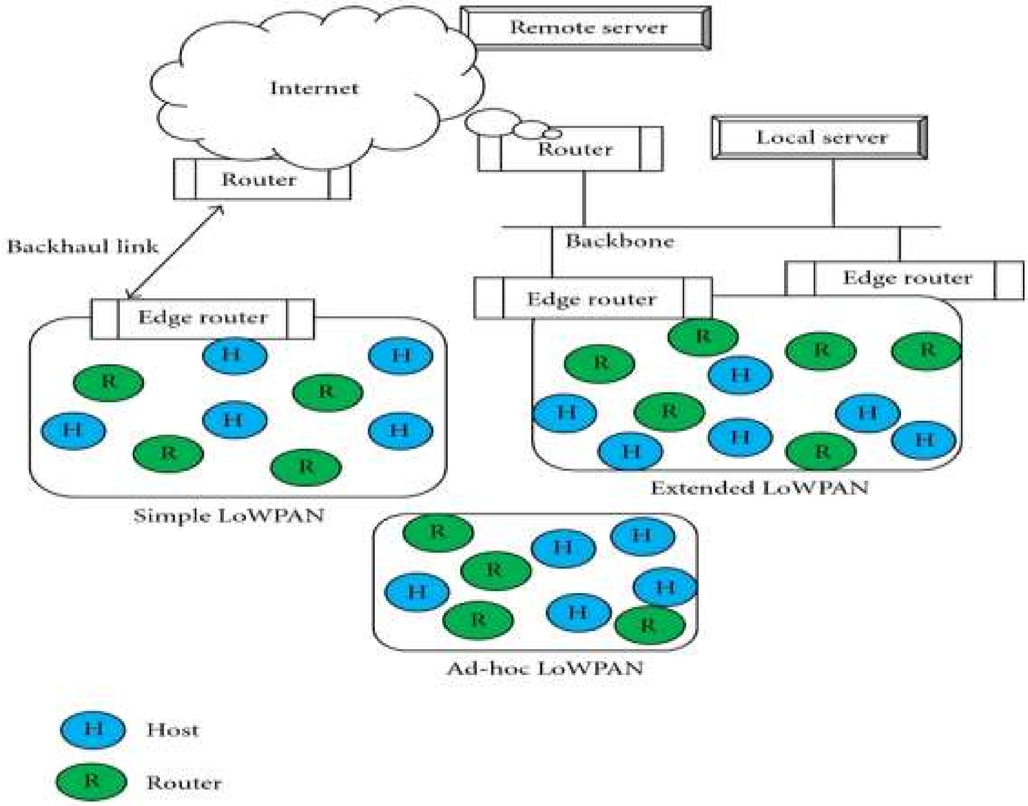


Figure 2- 13: 6LoWPAN Architecture (Al-Kaseem, Al-Dunainawi, & Al-Raweshidy, 2018)

2.7.2 Time-Slotted Channel Hopping (6TiSCH)

The IEEE 802.15.4e datalinks utilizes the TiSCH mode to allow IPv6 standard packets pass through it as developed by the IETF working group. It specifies available frequency selections and available timeslots used to schedule network operations in columns and rows of a matrix respectively. The time and frequency matrix have been designed in such a way that it will be known by participating nodes in the network, making nodes with overlapping interference ranges to agree on specific schedules.

Problems of scheduling often arises when nodes share same application and depending on the adoption by the MAC layer which can be centralized or decentralized (Dujovne et al., 2014). Similarly, scheduling is purely application specific as it is not specified by the standard thereby giving enough room for flexibility with different applications in IoT. It is self-governing with the ability to accommodate different network topologies and with a wide range of real-world application implementation. (Dujovne et al., 2014) explained the initial commercialization of TiSCH used in developing Time Synchronization Mesh Protocol (TSMP) which was comfortably used in Wireless HART networks as specified by (Specification, 2008). (De

Guglielmo, Brienza, & Anastasi, 2016) conducted a study with regards to the possible functional enhancements on TiSCH with emphasis on improvements on MAC performance metrics, low energy, enhancing beacons and increasing speed of association. (Qasem, 2018) summarized the enhancements done on IEEE 802.15.4e by (Vučinić et al., 2013) as follows; Asynchronous Multi-channel Adaptation (AMCA), Deterministic and Synchronous Multi-channel Extension (DSME) and Low Latency Deterministic Network (LLDN).

The combination of time-slotted access and channel hopping is believed to improve the reliability in communication through ease of access, improve energy efficiency allowing ultra-low power connectivity, increase network throughput and to reduce collision from contending nodes request for channel access which in turn reduces latency (De Guglielmo et al., 2016).

2.7.3 IPv6 Over LLNs (6Lo, G.9959 and Bluetooth Low Energy)

The focus on the development of standards by IETF 6Lo working group has been geared mostly to fashion out suitable datalink for the transmission of IPv6 frames for Resource-constrained nodes over the network. Some of the specifications defined by the ROLL working group for the transmission of IPv6 frames include, over IEEE 802.11ah, over DECT/ULE, over Bluetooth Low Energy Mesh, over Wireless Networks for Industrial Automation Process Automation (WIA-PA) (Gomez, Paradells, Bormann, & Crowcroft, 2017). The frame format defined for ITU-T G.9959 networks has an identifier which are unique for home network (32-bit) and host (8-bit) specified for each node. Frames are compressed by the 8-bit host identifier for IPv6 address to be constructed. Although security encryptions can be handled by the higher layer applications, a certain level of security has been defined by the RFC 7428 using a shared network key (Gomez et al., 2017).

However, Bluetooth Low Energy makes use of most compression techniques utilized by 6LoWPAN except for the fragmentation features, because enhancement of Bluetooth technology as defined by RFC 7668 already made necessary provisions for reassembly and segmentation of larger packets transmitted using its Logical Link Control and Adaptation Protocol (L2CAP). Bluetooth Low Energy uses a node acting as a coordinator/router for low-powered nodes to communicate effectively as multi-hop communication is not presently supported at its link layer (Nieminen et al., 2015).

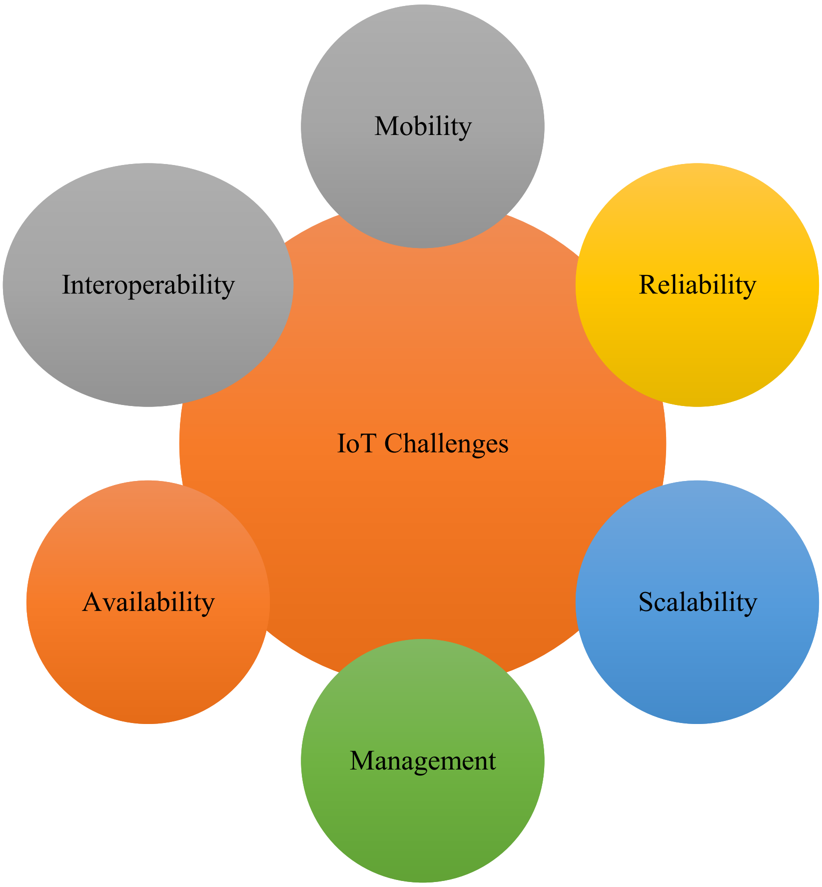


Figure 2- 14: Challenges of IoT

2.8 IoT Challenges

Despite the huge benefits of IoT, there are several challenges that hinders the successful development of IoT applications. Some of the challenges can be summarized in Figure 2-14.

Some of these challenges can be briefly explained below:

- **Reliability:** it entails ensuring that a system is working properly based on its design specifications. IoT requires data reliability especially in data collection, in communication and decision making.
- **Management:** Involves participating devices keeping records of performances, configurations and failures which is a major challenge in IoT considering the number of connected devices. Developers of IoT devices should be able to put into

consideration the acronym FCAPS which stands for fault tolerance, configuration, accounting, performance, and security.

- **Mobility:** Although RPL has not been designed to support mobility in IoT, such that it reconstructs new DODAGs each time a node moves out of the transmission range of its parent node or reconstruct the entire topology when devices change their gateways. Nevertheless, devices need to move freely which can result in changing both their IP address and in some cases change their service providers which increases additional complexities causing interruption to ongoing communications.
- **Interoperability:** Both homogenous and heterogenous devices should be able to interact seamlessly. There are different platforms for IoT implementation which makes interoperability very challenging, therefore manufacturers and application developers should be able to come up with both hardware and software that can work on any platform.
- **Scalability:** The projection on connected IoT devices is going to be in trillions in the next decade. Managing the highly distributive nature of such devices and their connections is going to be a serious challenge. Therefore, IoT platforms should be designed to include extensible service operations in other to accommodate new services and devices.
- **Availability:** Means services needed at any point in time should be provided. IoT applications for both software and hardware should be readily accessible by authorized individuals and compatible to different functionalities and protocols.

2.9 IoT Operating Systems

There are different operating systems used by IoT with each having different features both from hardware and software platforms. Although the most popularly used IoT OS in the academic environment is the Contiki OS (Qasem, 2018), with other common IoT OS such RIOT and Tiny OS.

2.9.1 Contiki

Contiki was first released in 2003 by Swedish Institute of Computer Science (SICS) as an open-source C-Language OS specifically for resource constrained applications (WSN and IoT), the project executed under networked embedded systems is a highly portable lightweight application that supports IPv4, IPv6, 6LoWPAN, RPL, CoAP and several other IoT protocol standards. It is divided into core and loaded programs (Qasem, 2018), with the core housing the communication stack, important kernel and libraries, drivers for devices, with different program loaders. Contiki uses protothreads which produces very low overhead in terms of concurrent programming with event-driven capabilities that only allocates memory to constrained devices on demand because of their inherent ability for low memory.

Protothreads operates very differently from the tradition multithreading programming concepts in that it doesn't need large memory to assign varying stacks for threads, it only assigns two bytes of memory for each of the protothreads (Reimers, 2012). Interfacing between programs in Contiki is achieved through a file known as Coffee, utilizing e-timer and r-timer for timing of events, and calling back of functions respectively. The recent Contiki version is the Contiki-NG (next generation) which was developed in 2017 in order to provide a more standardised communication in IPv6, to be more dependable, to accommodate more modernized platforms like ARM Cortex M3 and MCUs (with 32-bit Microcontroller Units) (ContikiNG, 2019), to provide better structure and configuration with better platforms, improvement in module description and code API, among several other improvements. The most recent is the Contiki-NG 4.3 developed in May 2019.

2.9.2 Tiny OS

Tiny OS is another operating system for constrained devices particularly WSNs which is developed in nesC an extension of C language for deeply networked systems having independent computational entities (commands, events, and tasks). It is event-driven developed at Berkeley having both inter-component and intra-component communication through events and tasks respectively (TinyOS, 2019). Command and event go hand in hand, as the former indicates a request initiated to perform a service while the latter indicates service completion. The design and implementation of Tiny OS has been motivated by the following points: firstly, limitation of resources, considering the size, cost and power consumption of the devices, secondly the reactive concurrency, this involves nodes having the ability to sample their environment with the aid of sensors, manipulating through actuators, processing, transmitting, routing, distributed processing and radio management. Thirdly, flexibility allowing application-specific and independence in terms of operation between hardware and software, lastly, low power in terms of battery design (Levis et al., 2005).

2.9.3 RIOT

RIOT is also another operating system which is both free and open source for WSNs and IoT. It is developed in C and C++ programming languages having the ability to support real-time applications and energy-efficiency with the inclusion of certain cryptographic libraries (Baccelli, Hahm, Gunes, Wahlisch, & Schmidt, 2013). It shows potentials of bridging the slit between operating systems for WSNs and full-fledged ones operating on internet hosts (Baccelli et al., 2013), it supports multi-threading, modularity and API access, having C++ implemented on it comes with Wiselib algorithm and TCP/IP network connection. It is developer friendly with a satisfactory level of reliability allowing developers to create all the necessary threads required as well as implementing distributed systems with the aid of an API. Modularity increases devices robustness against bugs where failure from one device driver or a file system does not affect the entire system. Scheduler is dependent on real-time support which signifies the ability to support fair distribution in terms of task priorities and user interactions. The scheduler in RIOT OS switch to idle mode in the absence of any event and only interrupts during wakeup thereby performing very little clock cycles.

Table 2-4 summarizes the most common comparisons of the OS used by IoT and outlining certain features which they share or differ with (Baccelli et al., 2013; Qasem, 2018).

Table 2- 3: Properties of Operating Systems in IoT (Qasem, 2018)

	Contiki	***Tiny OS***	***RIOT***
Modularity	*Partial*	*None*	*Full*
Real-Time Support	*Partial*	*None*	*Full*
Programming Model	*Event-Driven (Protothreads)*	*Event-Driven*	*Multi-threading*
Min RAM	*<2KB*	*<1KB*	*~1.5KB*
Min ROM	*<30KB*	*<4KB*	*~5KB*
MCU	*AVR®, MSP430™, ARM®, Cortex-M®, PIC32, Skymote/TelosB*	*AVR®, MSP430™, Skymote/TelosB*	*VR®, MSP430™, ARM®, Cortex-M®, x89.*
Memory Status	*Dynamic*	*Dynamic*	*Dynamic*
Protocols Supported	*RPL, COAP, uIPv4, uIPv6, , 6LowPAN, MAC IEEE802.15.4, TCP/ UDP.*	*RPL, 6LowPAN, COAP, MAC IEEE802.15.4, TCP.*	*uIPv6, 6LowPAN, MAC IEEE802.15.4, RPL.*
Scheduler	*Cooperative*	*Cooperative*	*Pre-emptive*
Language Supported	*C*	*nesC*	*C/C++*

Additionally, RPL routing protocol is one of the most utilized routing protocols in sensor networks, it is also implemented in several other operating systems such as LiteOS (which is a version of ZorinOS initially designed for use in calculators but was later adopted for low-resource computers such as sensor networks), T-Kernel developed in C++ language and EyeOS which allows for remote access and monitoring via the use of search engines.

2.10 Summary

This chapter outlines an in-depth overview of IoT devices, applications, layered protocol stack, and operating systems. It highlights the basic importance of optimizing the objective function within an RPL instanceID. Objective function often defines the metric consideration of an IoT routing protocol within the operating system, the default objective functions are Objective function Zero and Minimum rank with Hysteresis Objective function using hop count and expected transmission count respectively as metrics of consideration. This research recognises the works of previous scholars on the importance of using objective function to optimise the routing protocol in IoT, as well as highlighting processes and procedures of implementing the proposed objective function.

Chapter Three

Related Works

3.1 Introduction

Listing all the researched areas in IoT is an impossible task, but the aim of this chapter is to highlight all the key requirements and design implementations of previous works on IoT as well as presenting a broad understanding on progress and challenges of IoT. Similarly, it explains the importance of RPL routing protocol, which is the standard routing protocol for LLNs, and it outlines a systematic review of research insights and recommendations on RPL-based routing protocols in IoT. The review of related works was carried out from different research libraries with different research papers from google scholar, IEEE libraries, ACM, Elsevier, Springer, and several other libraries. The search returned thousands of papers, articles, journals, conference proceedings and patents where focus is been placed on most recent works or improvements of RPL routing protocol. Some of the key contributions of this section include, analysing the efficiencies of different approaches of RPL design based on flexibility, latency, packet delivery ratio and energy consumption, by conducting a systematic review. Also, it provides a guide to identify the technical improvements on RPL routing by other researchers.

There are numerous potential applications that utilizes IoT technology which can include, transportation, military, smart metering, home automation, agriculture, industries, and several other applications. Although some of the applications where studied extensively under WSN, some of the special requirements will be looked at from IoT perspective with the relevant examples.

3.2 Medical Healthcare

Medical healthcare presents a promising area where potentials of utilizing WSN and IoT are enormous. A great number of researchers explored the benefits and challenges of healthcare monitoring particularly in emergency detections, in monitoring vital signs for patients, Alzheimer’s elderly care monitoring, mental health care, among others. Medical healthcare is a very delicate aspect that requires a high-level functionality and performance from participating devices particularly in respect to reliability, mobility, responsiveness, availability, security and real time communication (Ko et al., 2010). Reliability is expressed in terms of providing accurate and timely information on patients, security is related to data protection and confidentiality of patient's information, responsiveness in areas that require immediate attention or in emergency situations, and mobility within an area or transmission area. Accuracy is another consideration in healthcare as explained in (Alves, Gabriel, de Oliveira, Margi, & dos Santos, 2015) whereby an inaccurate data for patient can lead to misdiagnosing or mistreating the patient by giving misleading outcomes. (Gao et al., 2008) conducted a research on using wireless medical sensor networks in emergency response with some implementations, the work used tags as sensors to determine the movement of patients in disaster scenarios, where part of the focus of the work was on obtaining reliable information from the tags in other to know the precise action to be taken at any given time.

Some of the actions include redirecting patients to the suitable hospital both in distance and medical provision, to dispatch ambulances accordingly, to provide patients medical history on-demand, etc. information reliability is also required in elderly care monitoring as discussed by (Chang et al., 2011), wearable sensors reliable fall detection (J. Chen, Kwong, Chang, Luk, & Bajcsy, 2006) and an ECG-recording system for wireless monitoring of tele-home care as proposed by (Fensli, Gunnarson, & Gundersen, 2005). Similarly, other healthcare research works like (Ren, Meng, & Chen, 2006) propose a wireless assistive sensor device for the deaf which does not require much reliability but is also another aspect of health monitoring using sensors. Monitoring of elderly patients is particularly challenging due to sudden fall, and this has been one of the main reasons of death in elderly care. Quick and timely intervention by medical healthcare personnel can save the lives of patients and prevent future occurrences. (De La Concepción, Morillo, Gonzalez-Abril, & Ramírez, 2014) and (Cruz et al., 2017) propose the use of accelerometers and use of camera with image sensors respectively to detect falls in a patient care unit. When the camera image sensor detects a fall, an emergency call will be made by the computer and RSSI will be used to determine the position of the patient in the building (H. D. Y. Kharrufa, 2018).

Usually, sensor devices deployed in healthcare are used to monitor and collect data on the habitual activities of the recorded entity including both the location and time duration. The sensor applications are mostly used with the intention of assisting medical staff or healthcare personnel in drawing recommendations. Other sensors can go as high as identifying symptoms of epilepsy and Parkinson's disease as discussed by (Patel et al., 2007) using wearable wireless sensors. The flexibility of certain routing protocols such as RPL allows different applications to be implemented by simply changing some configuration parameters to suit the design needs. More important research on medical care as discussed by (Huang, Wu, Huang, Wu, & Wang, 2021), with intensive research on elderly care as cited by (Tun, Madanian, & Mirza, 2021), and using IoT for chronic kidney disease as discussed by (Hosseinzadeh et al., 2021), are some of the most recent research on medical care (M. Javaid & Khan, 2021).

3.3 Smart Environments

Smart environment encompasses most of the automated applications in different aspects using WSN or IoT technology, they include smart buildings, smart metering, smart cities, smart agriculture, etc. The advantage of having the automated applications is to extend the reach of technology to certain areas bearing in mind the need for making such devices scalable and support mobility management. Smart environment includes almost all the automated services, but during this research some of the services provided by smart environment will be discussed individually as potential areas of research. It involves spreading sensors in different areas for different purposes to record and forward useful data such as humidity, temperature, pressure, etc which can be used in passive or active decision making through automated actions.

In smart agriculture, as discussed by (J.-A. Jiang et al., 2008) sensor nodes can be used detect crop or plant related diseases and mitigate the spread of the disease which is both beneficial to the environment and improves the economic advantage. In this regard, energy consumption and coverage area are very vital as sensors are expected to perform for a long duration of time while covering a very huge area. Some agriculturally related automated services include animal tracking and monitoring, which involves attaching sensor nodes to animals in other to record their migratory patterns, feeding habits, grazing areas, and communication coverage. Sensors can also be used to detect likely problems and diseases associated with animals as discussed by (Kwong et al., 2012) so that their productivity can be improved.

Similarly, (P. Jiang, Xia, He, & Wang, 2009) used a WSN to provide clean and safe drinking water for animals using wireless water monitoring system. Furthermore, some practical applications of smart environment have been implemented in some European cities which gathers data on the use of allocated resources and offers smart recommendations on the use of such resource (Casari et al., 2009). The likely challenges can be energy consumption and security considering the nature of deployment and services rendered, other environmental challenges that will likely affect the operational efficiencies and optimal performances of the sensors include strong winds, heavy rains, very low or very high temperatures. Robustness is a key concept for designing any smart environment related sensor network to overcome some of the challenges discussed.

3.4 Smart Transport

Smart transport is an important aspect that attracts significant research using WSN to monitor and collect data regarding traffic, congestion, faults, controlling traffic lights, emergencies, and traffic redirection. Most of these sensor applications has been deployed in other countries for different purposes, some are deployed to snapshot licence plates of traffic offenders, others detect emergency situations involving crash while others count the number of passing vehicles. With IoT technology implemented with these sensors, they could be able to do numerous things, for instance indicating an obstructing along the road and instantly placing emergency calls, detection of accurate lane positioning, initiating automatic breaks or more recently performing automatic parking (Qin et al., 2010; Tubaishat, Zhuang, Qi, & Shang, 2009). Similarly, sensors can be deployed in a more sophisticated manner like self-driving cars utilizing motion sensors, camera sensors and lane positioning sensors to drive the car as safely as possible.

Delay in information is one of most intolerable factors in designing these sensor systems to avoid life threatening situations, while factors like reliability and availability play key roles in supporting mobility for smart transport systems. Smart transport involves high-speed moving objects posing a great challenge of routing information considering the rapid changes in routing metrics. Security is also another challenge of smart transport application in life threatening situations due its vulnerability for cyber-attacks creating misleading information which will eventually lead to disastrous outcomes. Road and in-vehicle sensors are also branded as a segment of smart cities used for most traffic management plans for the design of future smart transportation. Suitability of deploying RPL routing for mobile vehicle to vehicle is not encouraging as compared to static road infrastructure sensor systems.

Research papers in RPL vehicular routing in IoT is not much, (Tian et al., 2013) depicted a Vehicular Ad Hoc Network (VANET) that predicts a vehicle’s direction by selecting a preferred parent within range using RPL routing protocol. The study as cited by (H. D. Y. Kharrufa, 2018) was only achieved by ignoring all aspects of energy conservation in RPL routing thereby removing energy saving as a consideration factor, and vehicles are assumed to be equipped with huge size batteries. Although the results are satisfactory in implementing on-road and in-vehicle deployment but there are arguments in using RPL to implement VANET scenarios and other mobility related services. More important research on smart transportation as discussed by (Munusamy et al., 2021), and a transport decision system as cited by (J. Li, Dai, Issakhov, Almojil, & Souri, 2021),and transport management system as discussed by (Bucchiarone, Battisti, Dias, & Feldman, 2021), are among the most recent research on smart transportation.

3.5 Industry Application

Industries have witnessed radical changes in automation systems and robotics by the continuous introduction of new technologies within the last decades, and it is one of the driving forces of automotive technological innovation. It achieves automation with the help of sensor actuators in form of control systems collecting, monitoring and acting on data around the surrounding vicinity. (Monshi & Mohammed, 2013) discussed some improvements in using WSN in power grid applications to form a smart power grid which is also another section of control system.

Smart power generation can improve efficiency in power distribution particularly with renewable energy applications such as solar energy or wind turbines when fused into the power grid, and it also simplifies the power generation process. Similarly, remote tracking of power usage and limitation of power wastage can be achieved using smart metering applications and remote sensing, thereby enabling convenience to users with huge economic benefits on one hand and transparency towards suppliers on the other hand. Using sensors can help to easily identify, locate and fix abnormalities around the surroundings, it can also be used in fire emergencies by identifying the fire location, automatically triggering the emergency plans in place like automated fire extinguishers or placing emergency calls to fire departments.

Industrial applications are designed and implemented putting certain metrics into consideration for optimal performance, these metrics include energy saving, minimum delay, and security. RPL can be a suitable routing protocol for industrial applications due to its reliability and flexibility in satisfying most industrial requirements. Some recent industry IoT applications on industry 4.0 as discussed by (Malik et al., 2021) and (Pivoto et al., 2021) are more recent research on industrial applications.

3.6 Military Application

Military application is another sensitive aspect where sensor technology is of paramount importance. It is particularly challenging due to the difficulty of deployment and assessment of physically placed devices, this makes it necessary to design military applications with energy saving been the most important metric consideration, as some situations present difficulty or high risk of replacing sensor batteries. The use of sensors for military applications are highly enormous ranging from providing emergency navigation to danger limitation towards troops with disaster prevention while using automated robots. Other uses of WSN in military applications include mine detection as proposed by (Niu et al., 2007) and the use of body sensors as wearable devices by troops to detect fatigues or health related issues as explained by (N. Javaid, Faisal, Khan, Nayab, & Zahid, 2013). Where energy saving is been considered as a key metric consideration, it works in tandem to mobility support, performance reliability and security in designing military applications. The support of active and passive data monitoring is highly required for accurate data generation to enable effective and reliable data transmission in military operations. (Qasem, 2018) simulated scenario using a SWAT robot vehicle, the vehicle gathers data and forwards it to intermediate sensor nodes through a gateway. More important research on military application as discussed by (Chippalkatti, 2021) on satellite based IoT for military applications, and (Ahmed, Nasr, Abdel-Mageid, & Aslan, 2021) on attack detection model, (Alsamhi et al., 2021) on green IoT using UAV on B5G networks and integration of IoT with drones as discussed by (Lakshman & Ebenezer, 2021), which are some of the most recent research on IoT network for military applications.

3.7 Received Signal Strength Indicator (RSSI)

RSSI is an estimated value of signal power that is received by a channel during transmission. It is an 8-bit integer value stored in the RSSI register and for newer radios such as CC2420 which is based on the IEEE 802.15.4 specification, it is determined within an 8-symbol period (128μs for 250kbps). Whenever the RSSI value is not found in the register meaning no transmission signal is detected, the value returned is determined as noise. According to (Bildea, 2013), RSSI range is between -28dBm to -127dBm, and RSSI can be affected by variation in the following, humidity (Lee & Chung, 2011), change in temperature by day or night (Boano, Tsiftes, Voigt, Brown, & Roedig, 2009), external interference causing increase in signal fluctuations (Zacharias, Newe, O'Keeffe, & Lewis, 2012), antenna alignment (Zhou, He, Krishnamurthy, & Stankovic, 2004), interference due to repeated transmissions (Mottola, Picco, Ceriotti, Gună, & Murphy, 2010), from hardware configuration (Tang, Wang, Huang, & Gu, 2007), above a certain threshold in CC2420 radios (Srinivasan & Levis, 2006), sensitivity to noise level (Srinivasan, Kazandjieva, Agarwal, & Levis, 2008), etc.

3.8 Challenges of RPL protocol

The most popular and one of the most widely researched routing protocols in LLNs is the RPL routing protocol. Although there are lots of research successes and enhancements made on RPL routing protocol and it is known for its flexibility, reliability, and energy efficiency, but there are also several routing challenges associated with it. These challenges will be discussed in detail within the coming sections.

3.8.1 Energy Consumption

Given the inherent nature of RPL routing is to minimize energy consumption in LLNs using trickle timer algorithm as it is one of the challenging issues bordering LLNs. Therefore, any design on sensor nodes must take energy consumption of nodes into consideration together with the minimization of control overheads. Although some researchers try to show that trickle timer performs poorly in dynamic situations which can cause packet loss and retransmission thereby consuming energy in the process.

Most RPL improvements have been on enhancing energy efficiency with most approaches using an objective function that uses energy metric in the design. According to a study by (Tsiftes, Eriksson, & Dunkels, 2010), RPL has an inherent energy saving nature where participating nodes can last for years and the argument was based on a simulation conducted where 40 packets were transmitted per minute. Additionally, (Lamaazi, Benamar, Imaduddin, & Jara, 2015) explained that energy consumption in nodes is directly proportional to the density and size of the networks, the believe is that nodes experience higher transmission and interference in denser networks. Similarly, a study carried out by (Shakya, Mani, & Crespi, 2017) in smart metering for industrial applications reveals 22%-27% increase in node lifetime using smart energy objective function (SEEOF) as compared to the traditional MRHOF default objective function for RPL. The SEEOF used residual energy as routing metric to be considered instead of the ETX and the results obtained were promising.

The study in (Mohamed & Mohamed, 2015) used a concept of ants to assume an independent participation of nodes in a network for carrying out decisions, this independence associate with nodes increases the welfare gain using remaining energy as the routing metric to increase network lifetime. Similarly, residual energy has been used as metric within the OF in most proposed routing protocols in efforts to ensure balanced energy distribution in the network to extend its lifetime. Other metric considerations from most research works that were used in the objective function include latency, throughput, loss rate, etc, which are all geared towards enhancing routing abilities of underlined routing protocols particularly in terms of reliability and availability.

Although both residual energy and energy consumption rates were considered as a routing metrics used to enhance RPL by most researchers in the past, it only forms part of the solution of a much bigger objective. Research on load balancing, congestion control and improved throughput will involve efficient energy distribution among nodes. In (Khan, Lodhi, Rehman, Khan, & Hussain, 2016), the authors used control message exchange in RPL to resize the network in relation to multiple sink nodes, while an energy consumption estimation model has been proposed to achieve improved energy distribution in the network (Banh et al., 2016). Similarly, (Zhen et al., 2020) proposed an enhanced energy efficient architecture for low earth orbit 6G internet of remote things with the aim of improving the energy consumption in random access schemes of satellite environments. (Behera et al., 2019) proposed a cluster head (CH) selection technique in internet of things based on residual energy of nodes and number of CH in the network for selection. Its uses a probability selection method with the aim of enhancing network lifetime. (Yaïci, Krishnamurthy, Entchev, & Longo, 2020) presented an overview of power and energy systems in internet of things, it discusses the importance of energy research in IoT. (Nesa & Banerjee, 2018) developed an algorithm for sensor data fusion that is energy efficient, they utilized SensorRank algorithm to dynamically determine network topologies and qualities of link before forwarding packets. (Ansere, Han, Liu, Peng, & Kamal, 2020) proposed an energy efficient cell free IoT using resource allocation by designing an optimized model for data transmission. (Kryszkiewicz, 2020) also proposed a neuron inspired communications for efficiency in IoT, it observes the human brains communication method to be utilized in IoT. (Tariq, Rehman, & Kim, 2020) focused on energy efficiency in software defined networks as a priority aware packet forwarding. (Qaim et al., 2020) wrote a review on energy efficiency in IoT highlighting important aspects of routing in wearable things and applying systematic literature on published papers from 2010 as well as comprehensive qualitative analysis. Other studies include routing and aggregation for minimum energy (RAME) by (Riker, Curado, & Monteiro, 2017) which uses least energy node information for traffic regulation and (Khelifi, Oteafy, Hassanein, & Youssef, 2015) which improves energy efficiency through fault detection.

This work presents a unique combination of different routing metrics to select an optimal path and preferred parent in an IoT network with particular emphasis on energy efficiency. Energy enhancements has been the most important aspect of IoT routing protocols, and this work have shown several improvements in different aspects of performance evaluations.

3.8.2 Mobility

Since the standardization of RPL routing in 2012 as a flexible routing protocol for LLNs, there were numerous efforts on investigating its viability for supporting mobility in IoT and mobile WSN. Its flexibility and scalability allow for interoperability solution for IoT applications utilizing RPL as a routing protocol. Several research efforts were made to utilize these functionalities to improve or develop better versions of RPL routing protocol. Although the inherent nature of RPL does not support mobility, but there is a mobility plugin and numerous researchers made efforts to provide solution for mobile support.

A study conducted by (Meddeb et al., 2018) developed an adaptive forwarding based link recovery known as AFIRM for IoT/Named data networking which is aimed at providing fail safe routes to disconnected routes in the network. It uses a combination of forwarding algorithms and data based-routing approaches to provide mobility support. (Gia, Rahmani, Westerlund, Liljeberg, & Tenhunen, 2018) developed a model for mobility support in IoT systems by leveraging positioning support and fog computing. Simulation results indicate an improved latency of 10%-50% as compared with other mobility systems in handover mechanism. Similarly, the use of reverse trickle timer algorithm in mobility simulation in RPL has been proposed by (Cobarzan, Montavont, & Noel, 2014) where child nodes were configured to contain a mobility flag and parent nodes are selected from the nodes without a mobility flag. The process is initiated when a mobile node becomes part of a DODAG, where the trickle timer is set to a maximum number while continuously reducing the value until it reaches the minimum, then node proceeds to a different parent. Some of the benefits of this approach is the reduction in disconnection and improvement in identifying unreachable parents. However, the downside of this method is that, different configuration settings were made to both mobile and static nodes thereby making the configuration difficult and time consuming.

(Liu, Zhang, & Zhang, 2013) designed a DAG-based multipath routing for mobile sensor networks (DMR) which uses link quality indicator and routing information as metrics. It is based on RPL routing protocol with a multipath approach providing alternative routes in case of route failure with a maintenance and repair mechanism in the DODAG. Although the authors showed that DMR performs better that mobile Ad hoc routing protocols (AODV and AOMDV), there was no comparison made with any mobility aware RPL protocol in this regard. (Saad et al., 2011) simulated a couple scenarios on WSNs using RPL in IPv6 to evaluate the load balancing and network throughput by deploying nodes with batteries and power line communication (PLC) acting as mobile sink nodes, the aim is to investigate the energy consumption of the nodes and this approach does not improve the performance of RPL. A similar approach for mobile sink in WSNs was investigated by (Saad & Tourancheau, 2011) combining and weighing three routing metrics hop-count, remaining energy and neighbour count, where the sink node chooses the path with most weight. This approach tends to increase network lifetime by balancing energy consumption across the network. A study on enhancing preferred parent selection on multiple mobile sinks using remaining energy metric for WSN proposed by (Safdar, Bashir, Hamid, Afzal, & Pyun, 2012), it is hybrid in nature and prevents nodes close to sink from running down by evenly distributing energy consumption thereby enhancing network lifetime. However, the approach has some disadvantages, firstly, the implementation design can only be achieved in the context where mobile sink is controlled and moved in a predefined manner, secondly, the only metric consideration is energy, and finally the concept was not validated by any practical implementation.

A proposal for the use of RPL in VANETs was made by (Tian et al., 2013), and the study focused on predicting nodes direction in adopting preferred parent by introducing a geographical information as a routing metric. The aim is to reduce the frequency of DODAG reformation and information dissociation by adjusting the DIO timer to be in line with node speed for effective handover, improved packet delivery and reduced latency. The downside is that the program uses a single cluster head which obtains data from static roadside sensors, it does not put network requirements into consideration and it further assumes that all nodes move in one direction.

(Tian et al., 2013) proposed the selection of preferred parent through the use DIO message header where mobile nodes status is indicated in the packet header, this way only static nodes will be selected as preferred parent. Simulated results indicate increased packet delivery and shows improved route stability, but due the changes applied to the DIO message header, it does not perform well with other RPL standards. Similarly, the use of reverse trickle timer algorithm in mobility simulation in RPL has been proposed by (Cobarzan et al., 2014) where child nodes were configured to contain a mobility flag and parent nodes are selected from the nodes without a mobility flag. The process is initiated when a mobile node becomes part of a DODAG, where the trickle timer is set to a maximum number while continuously reducing the value until it reaches the minimum, then node proceeds to a different parent. Some of the benefits of this approach is the reduction in disconnection and improvement in identifying unreachable parents. However, the downside of this method is that different configuration settings were made to both mobile and static nodes thereby making the configuration difficult and time consuming, secondly, it is built based on the assumption that mobile nodes are within the coverage area of a static node, and finally the absence of a mobility detection scheme.

A mobility aware model proposed by (El Korbi, Brahim, Adjih, & Saidane, 2012) where nodes sends ping messages when they move out of range of the preferred parents while the authors also introduced a destination searching method which sends continuous broadcast messages to detect nodes that are out of range. The mobility aware model is called “MoMoRo” which is only specific to low-power WSN applications, and simulation result shows similar packet delivery between traditional AODV and RPL, with more packet loss that RPL. The downside is that the protocol does not keep pace with high-speed nodes and simulation was done with few nodes making the process unrealistic.

In (Ko & Chang, 2014), the authors introduced a corona principle where the entire network is built in a circular manner around the DODAG root node, the aim is to ensure nodes locate alternative preferred parent quicker without reconstructing a new DODAG, and the second principle is the use of Fuzzy Objective Function FL-OF utilizing four routing metrics remaining energy, latency, hop-count and LQI. Performance evaluation of the protocol is better than the traditional RPL in PDR, energy saving and delay. Downside is that no mobility model was introduced, and low speed mobility of nodes were used. A concept utilized for the combination of mobile and static nodes developed by (Gaddour, Koubâa, & Abid, 2015) for healthcare and medical applications proposed a change in design of native RPL where mobile nodes do not act as routers or send DIO messages, but rather remain as child nodes without changing the objective function. The consideration of this approach is that when mobile nodes are selected as parents and suddenly it moves out of range, then the DODAG must be reconstructed all over again. There is an improvement in stability using this approach, but the downside is that it does not improve on native RPL rather simulates a specific scenario.

Furthermore, a study conducted by (Gara, Saad, Ayed, & Tourancheau, 2015) on the implementation of a mobility supported RPL for IoT applications hand-off time. The protocol known as mRPL aims improve the detection of disconnected nodes faster than previous approaches through the addition of four timers on the traditional trickle timer algorithm. Mobility of nodes in this approach is detected by the average RSSI while loss in connectivity is observed by the connectivity timer. The hand-off timer is configured in such a way as to operate in a short time bursts sufficient to send DIS and receive DIO thereby making further reduction in hand-off delay. Reply timer reduces collision by been responsible for replying mobile nodes and the mRPL performs better than traditional RPL in packet delivery, having less delay and overhead. The disadvantage is that this approach only considers average RSSI as the routing metric while ignoring link quality metrics thereby causing unnecessary and irregular handovers, secondly, only a single node is used to simulate the mobile scenario while moving at a very low speed and finally it shows no concern about the mobility management.

(Anand & Tahiliani, 2016) proposed mRPL++ putting mobility management into consideration with the aim of optimizing mobility in RPL. It is an improvement on mRPL such that nodes consider other link metrics aside from RSSI and objective function to select a preferred parent. Selecting a preferred parent can simply be based on the combination of average RSSI and metrics of the objective function. This approach inherits some of the downsides of mRPL in that it is RSSI dependant with little regard to the objective function. The authors in (Barcelo, Correa, Vicario, Morell, & Vilajosana, 2015) proposed a routing design based on position assistance in RPL for mobile and static sensors. It defines two modes which are static to static using ETX metric and mobile to static using KP-RPL with filters and blacklists. All communications to static nodes are listed by the mobile nodes according to RSSI value and it further blacklists those with very low ETX as having unreliable links. 25% network reliability has been achieved when simulation was done but only one mobile node is used in the simulation process without giving any projection on increasing the number of mobile nodes. Another downside is the fact that this approach blacklists nodes based on estimated positioning whereby any change in the position will degrade network performance, influence link reliability and routing decision.

In (H. Kharrufa, Al-Kashoash, Al-Nidawi, Mosquera, & Kemp, 2017), a D-RPL dynamic approach for routing in IoT applications has been proposed. It utilizes the concept of mRPL with regards to reverse-trickle timer in detecting mobile nodes while introducing additional metrics in the objective function to limit the frequency of unnecessary handovers to accommodate fast responsiveness. An extension of this approach using game theory (GTM-RPL) was conducted by (H. Kharrufa, Al-Kashoash, & Kemp, 2018) using metrics such as RSSI for mobile nodes, energy consumption and link quality level. This approach provides the best performance of RPL with mobility as it indicates significant improvement in throughput and energy consumption while providing solution that allows flexibility in the network.

3.8.3 Quality of Service (QoS)

QoS is geared towards achieving reliable data transmission which is bordered around improving data throughput, minimizing end-to-end delay and packet loss rate. The authors in (Dawans, Duquennoy, & Bonaventure, 2012) presented an approach to improve QoS in IoT applications based on the number of successfully received packets to determine the link quality in contrast to counting the number of sent control messages. It is a link quality consideration approach that ensures nodes only choose preferred parents based on it, and the simulation results shows an improvement in reliability and the maintenance of link quality values for directly connected nodes.

A cross layer design for link quality improvement in RPL was proposed by (Ancillotti, Bruno, Conti, Mingozzi, & Vallati, 2014) to utilize adaptive algorithm that aims to reduce energy consumption and latency with increased reliability of data transmission as compared with the traditional RPL routing protocol. A novel fuzzy logic based objective function was introduced by (Gaddour, Koubâa, Baccour, & Abid, 2014) which uses similar circular corona in (Ko & Chang, 2014) where nodes are formed in circular around the DODAG root node in other to be able to alternate between preferred parents based on four metrics hop-count, latency, remaining energy and link quality using FL-OF. Simulation results indicates reduced energy consumption, improved packet delivery and responsiveness with increased capacity to manage low speed mobility of nodes.

A similar fuzzy logic objective function approach to routing which uses a combination of metrics like remaining energy, latency and ETX was proposed by (P.-O. Kamgueu, Nataf, & Djotio, 2015), and simulation results also indicated an improvement in packet delivery, reduction in delay and energy consumption. In (Khelifi et al., 2015), a protocol known as Pro-RPL was proposed which can detect failed links by counting the number of packet loss and comparing it with the threshold. Each DIO message sent is encapsulated with information about energy and link cost which are used to make decision on preferred parent selection. Simulation results of this approach also indicates a slightly better performance as compared to traditional RPL protocol.

A study to detect the root node failure (RNFD) was proposed by (Iwanicki, 2016), where the authors dwell on the belief that root nodes do not always have enough power and can fail at any given time as opposed to most research papers. A probabilistic approach was utilized to detect preferred parent node failure covering huge network portion and root node failure. Drawbacks of this algorithm include additional energy consumption and overhead. In (Iova, Theoleyre, & Noel, 2015b), two routing metrics were combined by the authors which are remaining energy and ETX to create a multipath routing forwarding message through multiple preferred parents to root node. Simulation results indicate improved performance and energy efficiency as compared to traditional objective functions.

3.8.4 Congestion Control

Congestion can cause a lot of problems in the network, it can lead to increased energy consumption and end-to-end delay with reduction in network reliability and availability, making congestion control an important consideration in developing a routing algorithm. Similarly, in a multi-hop network, congestion increases with increasing number of nodes all transmitting very high data rates on a wireless channel. Some popular approaches to reducing congestion include control of transmitted traffic, hybrid schemes and resource control (H. D. Y. Kharrufa, 2018).

A Duty Cycle-Aware congestion control (DCCC6) as proposed by (Michopoulos, Guan, Oikonomou, & Phillips, 2012) for controlling congestion in 6LoWPAN. The DCCC6 uses RPL routing protocol having continuous congestion adjustment using buffer occupancy, simulation results indicate positive improvements in congestion control in RPL networks and other improvements in terms of reliability, availability, and energy consumption. Still on buffer occupancy, the authors in (Hellaoui & Koudil, 2015) proposed a congestion control mechanism used in testing bird flocking for RPL, 6LoWPAN and CoAP networks. The algorithm uses the buffer occupancy rate to select paths that are least congested and simulation results indicates improved performance in terms of congestion control in congested networks. (Al-Kashoash, Amer, Mihaylova, & Kemp, 2017) used different considerations in congestion control aside buffer occupancy and utility function for traffic and resource control, other metrics used in the study are queue delay, ETX, and multi attribute optimization.

In (Castellani, Rossi, & Zorzi, 2014) the authors discussed some schemes used to limit congesting in networks utilizing queue and buffer lengths or their combination, where simulation results indicate better performance in their combination than individual performance. In (Al-Kashoash, Hafeez, & Kemp, 2017), the authors introduced a game theoretic approach to congestion control where consideration is been made to energy consumption rate, both node and application priorities, and the buffer occupancy. Node and application needs were considerations ignored by (Castellani et al., 2014) by were heavily captured by (Al-Kashoash, Hafeez, et al., 2017), and simulation results indicates improvement in energy, throughput and reduced delay. The authors in (H.-S. Kim, Kim, Paek, & Bahk, 2016) used a queue utilization approach for load balancing and DIO messages exchanged between nodes in the network contains congestion information, and simulation results also indicated improvements in load balancing and congestion control.

In (Sheu, Hsu, & Ma, 2015), the authors proposed a preferred parent change approach using game theory where preferred parent informs leaf nodes of potential congestion in DIO message exchange, and simulation results indicates 100% improvement in terms of network throughput as compared to traditional RPL. Similarly, a load balancing approach utilizing multiple roots and multiple preferred parents has been proposed by (Ha, Kwon, Kim, & Kong, 2014), despite the recorded improvement in performance during simulation, it introduces more issues that do not conform to the standard operations of RPL.

3.8.5 Security

Security is a major concern in IoT applications considering the importance information been transmitted and their area of deployment. IoT applications are vulnerable to different types of attacks which can include exploitation, signal jamming or blockage, spoofing, denial of service or distributed denial of service, sink/black/worm hole attacks, etc, making the CIA (confidentiality, integrity and availability) of information a priority in IoT application. Security of IoT is outside the scope of this research and very little efforts were made by other researchers on the security of IoT. Most security architectures used are the ones provided by the standards in RFC 6550 which explains how the security in RPL is enabled and used. Three categories of security were standardised, unsecured (without any security measure), Pre-installed (a pre-installed key is required by all nodes joining the network) and authenticated (pre-installed nodes requesting operation as network routers).

A denial-of-service attack in the IoT applications prevents nodes from normal exchange of information causing continuous trickle timer to reset thereby forcing a loop in the DODAG and unnecessary repairs. These causes the energy of nodes to be depleted and network saturated with DIO messages. The solution adopted by (Hui, 2012) was to set a threshold for trickle timer resets within an hour which aims to minimize the energy expended in repairs and reformations. (Raza, Wallgren, & Voigt, 2013) proposed a solution for grey and black hole attacks using an intrusion detection system, and the algorithm tries to monitor end-to-end delays, DIO message exchange and loss packets. Their simulation results indicate positive outcomes in detecting and preventing attacks from malicious nodes.

A study by (Mayzaud, Badonnel, & Chrisment, 2017) introduced a method of detecting malicious activity by monitoring the network. It is built on the belief that nodes cannot accept cryptographic messages and still perform as member nodes in the network, but rather different nodes should be deployed, some for message exchange and the others for monitoring the network. Simulation results indicate that the approach prevents malicious activities in the network but create more overhead from the monitoring nodes added in the network.

3.8.6 Multipath Approach

Multipath routing approach has been in research for decades with the aim of creating multiple paths to destination and for other reasons which include designing a system that is fault tolerant, better QoS and overall load balancing performance (Barthel et al., 2012). RPL uses a directed acyclic graph (DAG) to avoid loops and create different paths to destination thereby making multipath research have little significance, but it is been considered in most research related to WSNs.

A study on multi-parent routing for balancing the energy in RPL using a combination of traffic load with quality of the link to produce a new metric known as expected lifetime (ELT) was proposed by (Iova, Theoleyre, & Noel, 2015a), the simulation used constrained devices and was conducted and evaluated in WSNet simulator. Similarly, (Pavković, Theoleyre, & Duda, 2011) proposed a multipath opportunistic RPL routing in IEEE 802.15.4 with QoS support and energy saving as the target, simulation results for the algorithm indicates improved packet delivery and reduced latency. In (Le, Ngo-Quynh, & Magedanz, 2014), a slightly different routing approach to standard RPL has been introduced where preferred parents are selected from nodes having the same rank number as opposed to the normal parent selection of lowered number ranks. The algorithm used a combination of remaining energy and hop-count for achieving load balancing and simulations were made in OMNET++ simulator with results indicating slight improvements, but the overhead generated from switching parents are not worth the improvements.

In (Zhu, Wang, & Yang, 2017), a multipath data distribution mechanism based on RPL for energy consumption and delay has been proposed where the algorithm focuses on balancing energy in participating nodes to increase the network lifetime. The study used a different approach to preferred parent selection using a probability mitigating buffer overload using cache utilization for multipath forwarding in the network. Simulation results indicate improvements in network lifetime and delay, but load balancing has not been achieved for DODAG root directly connected nodes. One hop neighbour to DODAG root are important because they serve as a gateway for other nodes to connect to the DODAG root. Additionally, a parent aware OF (PAOF) was proposed by (Gozuacik & Oktug, 2015) which combines two metrics, parent count and expected transmission count (ETX) in the selection of preferred route to DODAG root, simulations and evaluations were done in COOJA simulator with results indicating positive performance as compared to the default routing in RPL.

3.8.7 MAC (Queue Control) Approach

Medium access control (MAC) layer protocols in constrained networks is also responsible for scheduling of channel access and sharing of communication medium, which as highlighted by most researchers in this aspect that congestion in the network can lead to overloading parent nodes creating buffer limitation and unnecessary energy consumption within the RPL network. A study aimed at reducing end delay and load balancing was proposed by (H.-S. Kim et al., 2016) using queue utilization metric to select the preferred parent in tandem with hop-count.

Investigations on implementation indicates improved packet delivery and reduced delay, while the energy consumption aspect of the research was not analysed.

In (Chekka, Miao, & Kim, 2014), an adaptive binary exponential back off (ABEF) algorithm that determines buffer overflow while utilizing several parameters was proposed. It indicates better performance regarding packet drop as compared to traditional protocols but like in (H.-S. Kim et al., 2016) the power consumption and link quality aspect in nodes has not been investigated. Similarly, in (Di Marco, Athanasiou, Mekikis, & Fischione, 2016) a cross-layer mathematical model concept which explained a dynamic interaction of MAC IEEE 802.15.4 working on RPL layers in lossy networks, taking the dynamic behaviour of medium access control while relying on contention to improve the transmission count reliability, network lifetime and load balancing functions. The model introduced in the MAC layer are R-metric and Q-metric, the first metric extends ETX to include loss packets used to measure end delays, while the latter metric measures the level of contention and ignoring the queues.

The authors in (Tall, Chalhoub, & Misson, 2017) proposed a multichannel collaborative load balancing algorithm with queue overflow avoidance in wireless sensor networks which is geared towards improving throughput by aiming to minimize collision, congestion and interference. Minimizing the queue overflow also means minimizing the loss packets to achieve better load balancing, where simulation results indicate improved performance as compared to traditional RPL and as compared to above papers energy consumption analysis was neglected.

3.9 Routing Metric Awareness in RPL

Within any routing protocol, there are metric information that can be utilized to make routing decisions for different application needs that can be either in real-time or emergency situations. Such routing metrics can also be defined as constraints in making routing decisions, and it is a quantitative value which is used to evaluate and determine a path cost. The most suitable path in routing is the one that best satisfies all constraints and simultaneously has the least cost according to some defined metrics. Routing metrics can be categorized as dynamic/static, link/node and qualitative/quantitative (Barthel et al., 2012) as defined in the objective function, and routing can either use a single metric or a combination of several metrics (composite metric).

3.9.1 Single Metric

The default routing Objective Functions (OF) in RPL all use the single metric (OF0 and MRHOF) to determine routes in the DODAG, and this concept performs better in reducing the level of complexity in the OF. However, by automatically neglecting other important metric considerations, the destination oriented directed acyclic graph construction can never be fully optimal in its operations. Using a single metric is inadequate and does not guarantee the satisfaction of different application needs, an example is in the case of trying to save energy for a particular application and it happens that the underline routing objective function is hop-count, in this case the algorithm will continue determining shortest path without necessary giving concern about the energy consumption thereby causing the batteries to run out causing total network failures. As stated by (Qasem, Al-Dubai, Romdhani, Ghaleb, & Gharibi, 2016) and (H.-S. Kim et al., 2016), choosing an objective function that relies only on ETX as the routing metric increases the end-to-end traffic due to unbalanced load traffic, without doubts this creates issues of reliability and latency in real time applications of IoT.

3.9.2 Composite Metric

Several researchers made efforts to optimize the performance of RPL by trying to combine two or more metrics in the objective function. According to (Pavkovic, 2012), most approaches requiring the combination of two or more metrics should be achieved either through hierarchical/lexical concatenation, linear/additive combination or fuzzy logic.

3.9.2.1 Lexical/Hierarchical Approach

This approach of combining metrics in RPL routing is beneficial especially if the metrics to be considered are obtained from a range of values as opposed to using a particular value. When two or metrics in this approach happen to have the same metric value, a secondary metric is always triggered to be a tie breaker in the selection of the preferred parent. These problems rarely happen, and some researchers discourage this approach in proposing the optimization of RPL performance (Qasem, 2018), but a lexical combination of metrics based on a range of values without necessarily maintaining the order of routing metrics could be a viable solution to this problem.

A lexical combination of hop-count and packet forwarding indication (PFI) metrics proposed by (Karkazis et al., 2012) designed an algorithm for primary and composite routing metric for

RPL-compliant WSN, it removes maliciously misbehaving nodes while building shortest paths in the network. Similarly, (Abreu, Ricardo, & Mendes, 2014) proposed an energy-aware routing for biomedical wireless sensor networks using a combination of remaining energy and ETX as the OF. The similarity in the ETX value causes the remaining energy aspect to be activated, making the selection of the preferred parent to be based on the parent with the most remaining energy value. Simulation results indicate improved network lifetime and energy saving.

A cyber adaptive physical objective function has been proposed by (Ghazi Amor, Koubâa, Tovar, & Khalgui, 2016) for use on smart cities, where DIO packet has been equipped with an alarm that notifies the root node for any events, then the root node triggers a new DODAG construction that best fits the pending event. This approach tends to consume more energy and cause more delay due to DODAG reconstruction been event-based, and reliability is a concern especially when it involves huge number of nodes. Similarly, a context-aware addressing scheme for RPL networks was proposed by (Kalmar, Vida, & Maliosz, 2015) for data-centric communication where RPL trees were used for aggregating contextual decision information.

3.9.2.2 Additive/Linear Approach

This approach involves the continuous addition of metrics which are calculated as a form of one metric. The order of routing metrics in approach is very important unlike the lexical method, and parameter or metrics are arranged based on different application requirements. In (Nassiri, Boujari, & Azhari, 2015), the authors proposed an energy-aware parent selection in RPL for load balancing in WSN where four routing metrics were combined in both additive and lexical manner, the metrics are ETX, hop-count, remaining energy and RSSI. The study emphasises on the possibility of combining several metrics and using composition evaluation function on DIO packets. In the additive approach, metrics are attached to different weights depending on their level of importance and the resultant added to produce a single value. Combination of several metrics tend to optimize performance of RPL, but it also affects the performance of the system, this makes the necessity for metric combination to always echo on application needs.

A fault-tolerant RPL for context awareness has been proposed by (Sharkawy, Khattab, & Elsayed, 2014) which uses remaining energy to improve the power consumption of nodes at both active and sleeping modes. The study builds a multicriteria prediction method utilizing approaches in (Musolesi & Mascolo, 2008) and (Mascolo & Musolesi, 2006) respectively which takes into cognisance the remaining energy of nodes and duty cycle to calculate rank and preferred parent in an additive manner. Similarly, remaining energy and queue utilization were used by (Taghizadeh, Bobarshad, & Elbiaze, 2018) to come up with a context-aware routing for load balancing in heavy and highly dynamic load. Simulation results were promising but other metrics that would affect the performance of such network were not considered.

3.9.2.3 Fuzzy Logic

The concept of fuzzy logic involves the combination of different parameters to create membership functions using a probability distribution to mimic the human reasoning which is mostly considered erratic and flexible. It uses algebra-reasoning to transform multiple input variables into a single qualitative output. There are several researches on utilizing fuzzy logic for wireless sensor networks in different applications as well as optimizing the objective function (Gaddour et al., 2015; P.-O. Kamgueu et al., 2015) which could either be based on two or more metric combination in selecting the preferred parent or best route.

Both user and application requirements play an important role in the selection of a suitable objective function and according to different reviews from previous research works, there is a disadvantage in using a single objective function. For instance, a preferred parent might be chosen as node having the lowest ETX, but that parent might be consuming more energy than other alternative parent nodes. In (P.-O. Kamgueu et al., 2015), the authors used a fuzzy logic model to combine several routing metrics in efforts to optimize the objective function. The proposed OF was compared to the default OF where simulation results indicate improved performance in terms of energy consumption and PRR. Although these performance metrics are directly proportional to the data traffic and number of nodes, efficiency was not guaranteed as comparison was done by single metric objective function only. The metrics combined were end-to-end delay, ETX and residual energy.

Similarly, in (Gaddour et al., 2015) a new fuzzy logic objective function was proposed, the objective function combines four link and node metrics to make routing decisions while using fuzzy parameter configuration. Simulation results indicate improved performance in terms packet loss, latency and network lifetime as compared to native RPL OF. The metrics used are remaining energy, hop-count, ETX and latency. In (P. O. Kamgueu, Nataf, Ndié, & Festor, 2013), another objective function was proposed that uses a single metric (residual energy) for path calculation which is built on the concept of MRHOF but using a node metric instead of a link metric. Simulation results indicate improved energy distribution and network lifetime with a high level of transmission accuracy.

In (Agarkhed & Kadrolli, 2017), the authors proposed fuzzy logic in cluster head selection in hierarchical routing while dividing the cluster into sub-clusters. Simulation results indicate improved performance in terms of energy consumption and network lifetime as compared to LEACH and CHEF routing protocols of WSN, but the protocol used only one cluster head without the consideration of multiple cluster heads. Similarly, the authors in (Sharkawy et al., 2014) introduced context-aware optimized objective function, which considers the temporal changes of preferred parents in sensor nodes and the limitation of their resources to base its fundamental consideration on remaining energy for making routing decisions. Simulation results indicate 44% increase in network lifetime, with increased packet delivery while ensuring fairness as compared to other non-contextual objective functions of RPL. However, network traffic consideration was ignored by the study as well as the effectiveness of the protocol in a mobile environment.

In (Nurmio, Nigussie, & Poellabauer, 2015), two parent objective functions (PEOF, PEOF2) were proposed for equalised energy distribution among member nodes to improve the network lifetime. PEOF combines residual energy with ETX as routing metrics to choose one-hop preferred parent while PEOF2 residual energy along the preferred parent paths with each hop. Simulation results of both proposed approaches indicate better performance in terms of energy consumption as compared to default RPL OF while all OFs show similar packet delivery, and performance was analysed in a symmetric and asymmetric network.

Similarly, in (Shakya et al., 2017), the authors introduced an OF that focuses on smart energy efficiency to prolong network lifetime using a combination of two metrics energy level and ETX. Simulation results indicate 27% network lifetime improvement with uniform energy consumption as compared to default RPL OF. Additionally, an OF that combines hop-count, packet forward indicator (PFI), residual energy and ETX was proposed by (Yunis & Dujovne, 2014). Simulation results were investigated which indicates that the combination of residual energy and hop-count show energy saving without additional packet loss or delay. The authors further explained the significance of each routing metric in DODAG construction, hop-count was able to minimize delay and packet loss, PFI ensures the survival of nodes where loss nodes are observed when preferred parents do not retransmit.

In (Ghataoura, Yang, & Matich, 2009), a genetic adaptive fuzzy hop selection scheme was proposed by the authors to select the best route in robust networks of WSN routing using different channel conditions. It uses a fuzzy system with SNR and network outage as inputs in selection of preferred neighbour node. Simulation results as compared to crisp approach indicate improvements in energy consumption and reliability at 15% and 20% respectively.

In (Gupta, Riordan, & Sampalli, 2005), the authors proposed a fuzzy based cluster head selection for hierarchical routing aimed at improving the downsides of LEACH protocol. The concept of LEACH allows cluster heads to be selected based on local arrangements and a stochastic probability model, especially when nodes are located far from each other or very close thereby unnecessarily maximizing energy usage. Simulation results indicate improved network lifetime using fuzzy variables.

Similarly, the authors in (Zeynali, Khanli, & Mollanejad, 2009) proposed a concept of using distance and remaining energy metrics in a fuzzy spanning tree for cluster head selection in hierarchical routing. The combination of these two metrics produces a fuzzy election number that guides route and preferred parent selection, simultaneously balancing energy consumption among nodes. However, the authors in (ALMomani & Saadeh, 2011) used a slightly different approach in hierarchical routing, where the logical tree topology is built by the protocol not by clustering mechanisms. Fuzzy inference is used for ranking nodes making neighbours rank to be based on their energy consumption level and depth. Simulation results indicate higher overhead construction.

A fuzzy logic system can be represented using the four units as explained by (Qasem, 2018), fuzzifier, inference engine, defuzzifier and a knowledge base, and it can be represented graphically as in Figure 3-1. Using instinct and judgment play a crucial role in developing fuzzy logic to computer models, also with the integration of fuzzy sets, operations, relationships, inference engines, membership functions, rule-based database, fuzzification and defuzzification.

The following points will highlight the use of FL in developing the adaptive objective function as implemented in this thesis. Membership function tends to describe the vagueness of boundaries and describing the relationship of an element drawn in a fuzzy set while representing it in a graphical form (Zuo, 2013).

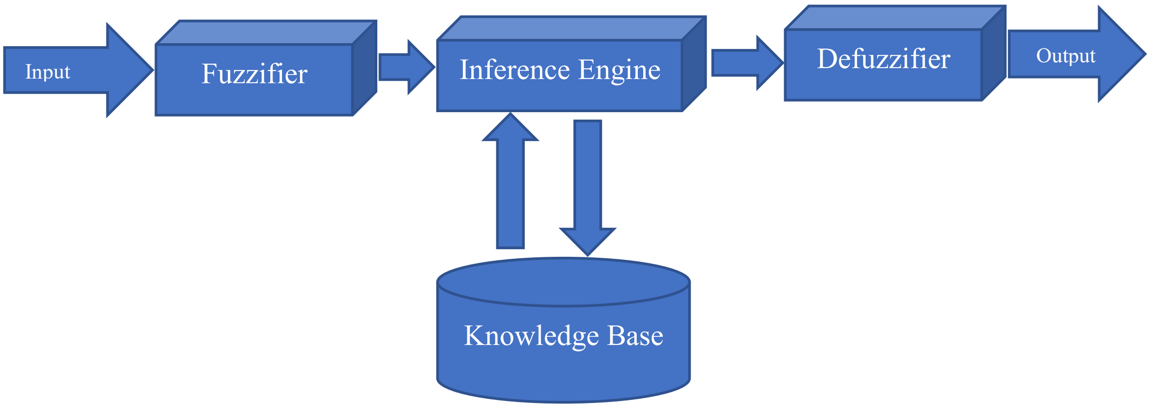


Figure 3- 1: Fuzzy Logic System (Qasem, 2018)

The following equation has been explained by (Zuo, 2013);

Let $\underline{D}$ be a fuzzy set, with an element belonging to the membership function of the fuzzy set as x, such that the fuzzy set $\underline{D}$ is indicated as $\varpi_{\underline{D}}(x)$ where $\varpi_{\underline{D}}(x) \in [0,1]$, then $\underline{D}$ could be defined in equation 2.1:

$$\underline{D} = \left\{\frac{\varpi_{\underline{D}}(x_1)}{x_1} + \frac{\varpi_{\underline{D}}(x_2)}{x_2} + \frac{\varpi_{\underline{D}}(x_3)}{x_3} + \frac{\varpi_{\underline{D}}(x_4)}{x_4} + \dots\right\} = \left\{\sum_i \frac{\varpi_{\underline{D}}(x_i)}{x_i}\right\} \qquad \mathbf{3.1}$$

Alternatively, the above equation can also be represented as in equation 2.2,

$$\underline{D} = \left\{\int \frac{\varpi_{\underline{D}}(x)}{x}\, dx\right\} \qquad \mathbf{3.2}$$

Aside from some exceptions made by certain researchers, most valid operations existing in classic element sets is applicable in this regard. Fuzzy sets are known for their ability to overlap, hence the combination of either the complement or the union will not include the whole set of x.

Using fuzzy relations to explain the association between two elements on universes (X and Y) by cartesian product, defining a set $\underline{D}$ in universe X and set $\underline{E}$ in universe Y whereby the cartesian product of the two sets is $\underline{D} \times \underline{E} = \underline{F}$, and the resultant relation $\underline{F}$ formed on cartesian space with product of the two universe $X \times Y$ is within the interval [0, 1]. The formation is categorized by the membership function from the resultant $\underline{F}$ using chosen pair of ordered elements from the universes $\varpi\underline{F}(x, y)$. Therefore, the resultant membership function is given by the fuzzy relation as in equation 2.3:

$$\varpi_{\underline{F}}(x,y) = \varpi_{\underline{D}\times\underline{E}}(x,y) = min[\varpi_{\underline{D}}(x), \varpi_{\underline{D}}(y)] \qquad \mathbf{3.3}$$

The above max-min and product form the common composition for membership function operations. Similarly, defining another fuzzy relation of $\underline{F},\ \underline{G}\ and\ \underline{H}$ with products as $X \times Y$, $Y \times Z$ and $X \times Z$ respectively on a cartesian space. Therefore, the common composition of the theoretic membership functions can be expressed in equations 2.4 and 2.5:

$$\varpi_{\underline{H}}(x,z) = \bigvee_{y\in Y}[\varpi_{\underline{F}}(x,y) \wedge \varpi_{\underline{G}}(y,z)] \qquad \mathbf{3.4}$$

$$\varpi_{\underline{H}}(x,z) = \bigvee_{y\in Y}[\varpi_{\underline{F}}(x,y) \cdot \varpi_{\underline{G}}(y,z)] \qquad \mathbf{3.5}$$

Where the min, max and product of elements are expressed as $\wedge, \vee\ and\ \cdot$ respectively.

Membership function of a set contains all the necessary fuzziness required with all the required shapes such centroid, triangle, cosine, trapezoid etc. Fuzzification entails passing on arithmetical input to fuzzy sets with varying degree of membership ranging between [0,1], where 0 signifies that the value is not in fuzzy set and 1 indicates otherwise, whereas values that fall within 0 and 1 indicates degree of uncertainty of whether the value belongs to the set or not. Fuzzy set can also be assigned words as system input so that the reasoning can be completely done in a linguistical manner. Some other methods of assigning values to variables in membership sets as genetic algorithms, inference, neural networks, etc.

Whereas defuzzification entails getting quantifiable results in crisp logic which involves mapping a fuzzy set with the required crisp set, meaning that the output produced by the rule-based system could be a logic union of already defined membership functions expressed as a max operator represented by *U*. The following four commonly used defuzzification approaches where the crisp output is indicated as x^* are given in Figures 3-2 to 3-5. Fuzzy logic establishes a leeway to classical logic of reasoning under uncertain situations where partial or approximate truths are utilized to include an extension of the well-known Boolean logic of true or false outcomes to accommodate multi-valued logical reasoning representing varying degrees of truth.

Centroid:

$$x^* = \int \varpi_{\underline{D}}(x) \cdot x dx \Big/ \int \varpi_{\underline{D}}(x) dx \qquad \mathbf{3.6}$$

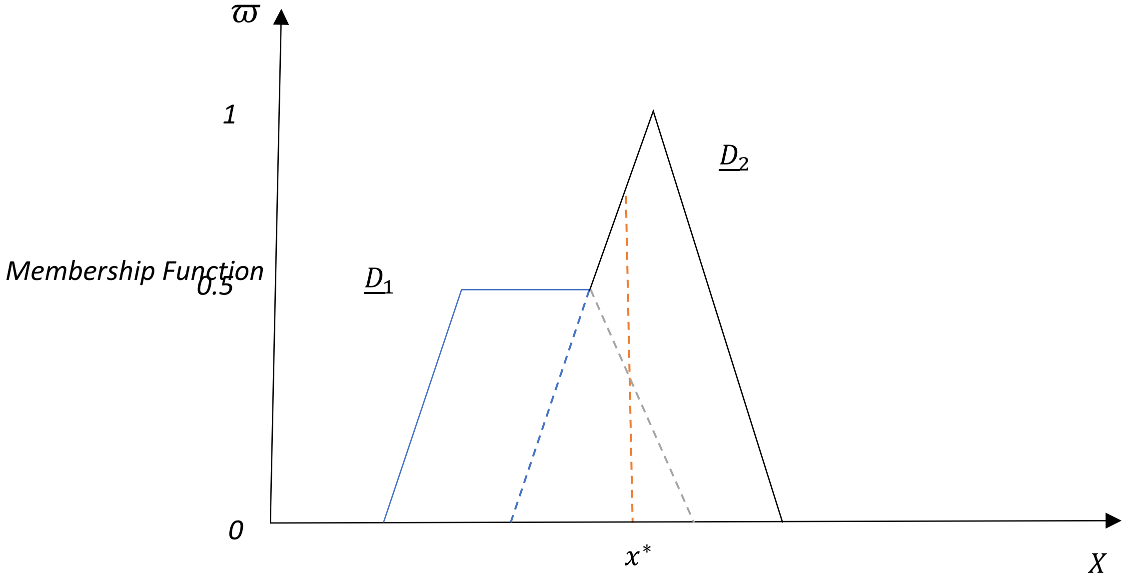


Figure 3- 2: Centroid Method (Zuo, 2013)

Max-membership:

$$\varpi_{\underline{D}}(x^*) \geq \varpi_{\underline{D}}(x) \qquad \mathbf{3.7}$$

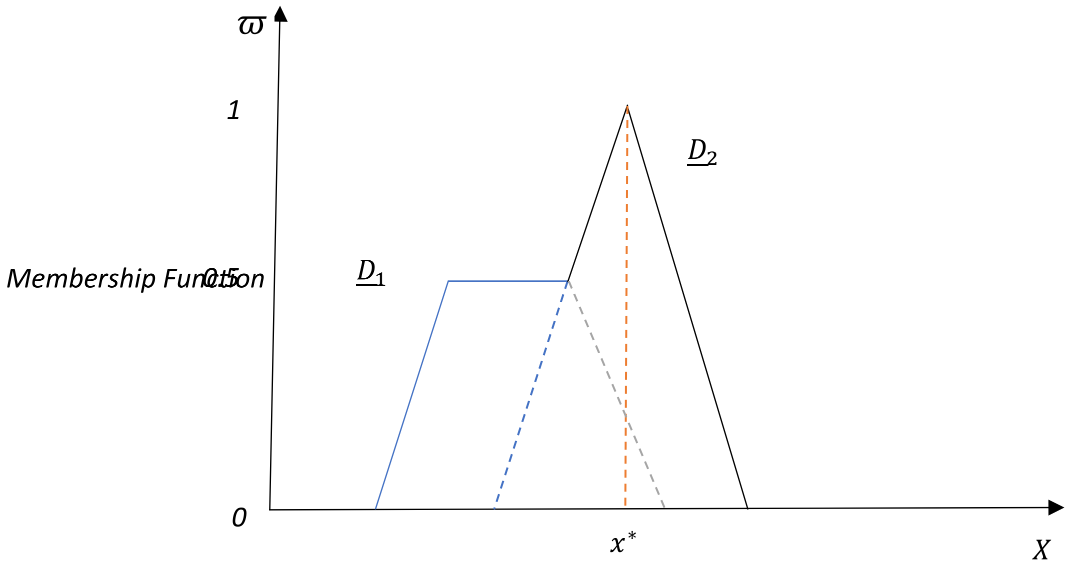


Figure 3- 3: Max-membership Method (Zuo, 2013)

Weighted Average:

$$x^* = \sum \varpi_{\underline{D}}(\bar{x}) \cdot \bar{x} \;/\; \sum \varpi_{\underline{D}}(\bar{x}) \tag{3.8}$$

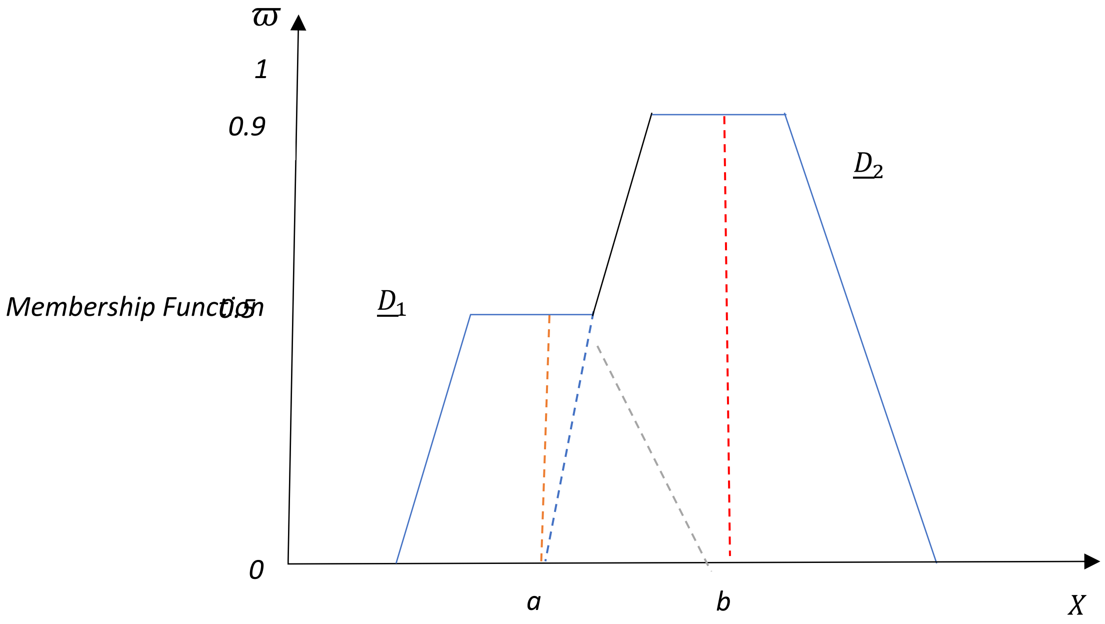


Figure 3- 4: Weighted Average Method (Zuo, 2013)

Mean-max membership:

$$x^* = (x' + x'') \;/\; 2 \tag{3.9}$$

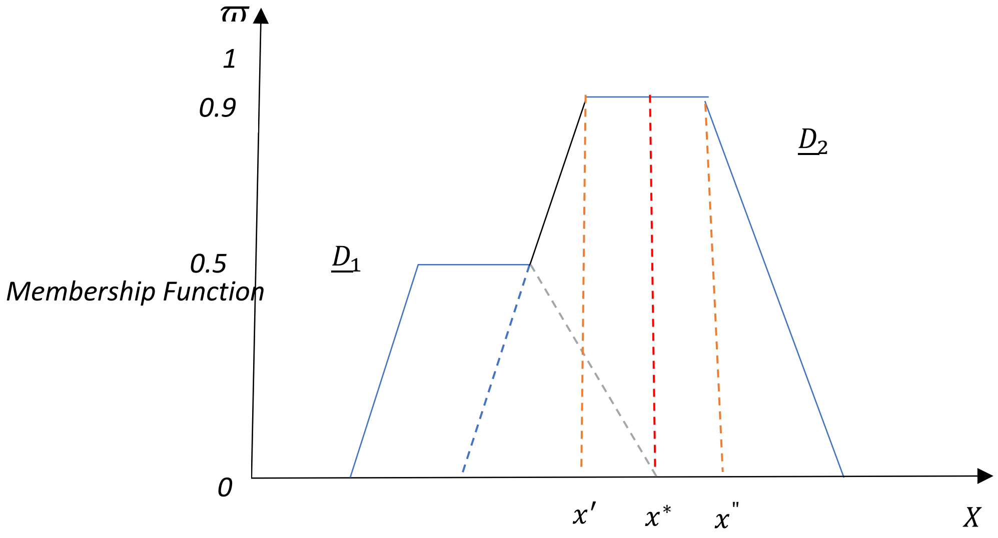


Figure 3- 5: Mean-max Membership Method (Zuo, 2013)

Assuming the proposition $\underline{K}$ will be assigned to set $\underline{D}$, therefore the value of truth detonated as T ($\underline{K}$) must satisfy equation 2.10:

$$T\ (\underline{K})\ =\ \varpi_{\underline{D}}(x)\ \ where\ 0\ \leq\ \varpi_{\underline{D}}\ \leq\ 1 \qquad \mathbf{3.10}$$

The degree of truth can be seen from the equation above, where proposition $\underline{K}$: $x \in \underline{D}$ represents the membership value of x in set $\underline{D}$. logical connections between any two propositions can be defined as follows: ($\underline{K}$ *and* $\underline{J}$)

- **For Negation:** which is unary logical complement of a proposition of an element

$$T\ (\overline{\underline{K}})\ =\ 1\ -\ T\ (\underline{K}) \qquad \mathbf{3.11}$$

- **For Disjunction:** is an alternation or a truth-functional operation where the combination of the set of operands are true if any of the operands are true or both.

$$T\ (\underline{K}\ \vee\ \underline{J})\ =\ max\ [T\ (\underline{K}), T\ (\underline{J})]\ (x\ is\ \underline{D}\ or\ \underline{E}) \qquad \mathbf{3.12}$$

- **For Conjunction:** is a truth-functional operation where both operands must be true

$$T\ (\underline{K}\ \wedge\ \underline{J})\ =\ min\ [T\ (\underline{K}), T\ (\underline{J})]\ (x\ is\ \underline{D}\ and\ \underline{E}) \qquad \mathbf{3.13}$$

- **For Implication:** is a logical connective term used to form conditional statements, meaning one operand is the cause while the other operand is the consequence,

$$T\ (\underline{K} \rightarrow \underline{J}) = T\ (\overline{\underline{K}} \vee \underline{J}) = max\ [T\ (\overline{\underline{K}}), T\ (\underline{J})] \qquad \mathbf{3.14}$$

The propositions $\underline{K}$ *and* $\underline{J}$ are defined on sets $\underline{D}$ *and* $\underline{E}$ respectively, where a fuzzy relation based on set theory approach will be $\underline{M} = (\underline{D} \times \underline{E}) \cup (\overline{\underline{D}} \times Y)$, and Y represents universe of set $\underline{E}$. $\underline{M}$ membership function is represented in antecedent-consequent format,

$$\varpi_{\underline{M}(x,y)} = max\ [\ (\varpi_{\underline{D}\ (x)} \wedge \varpi_{\underline{E}\ (y)}), \quad (1 - \varpi_{\underline{D}\ (x)})] \qquad \mathbf{3.15}$$

As discussed in this section, the use of fuzzy method is very vital in combining different routing metrics which allows certain input variables to pass through the fuzzification process, inference engine and rule-based system to produce a single optimal output via the defuzzification process. A complete implementation of this FL OF is given in chapter 7 of this work. Table 3-1 can be summarized in Figure 3-6 which outlines all the network simulators utilized in this research. The 'others' category represents a group of unpopular simulation models used for IoT applications, most of which were originally designed for WSN simulation, they include NS3, NS2, TestBed and TinyOS with MoMoRo.

3.10 Summary of Related Works

Table 3- 1: Summary of Cited Works in Literature (Halilu & Hope, 2021)

Research Papers	Energy	Mobility	QoS	Congestion Control	Security	Multipath	MAC (Queue)	Implementation Environment	Metric/ Strategy	Approach			Comments
										Hierarchical	Linear	Fuzzy	
(Halilu & Hope, 2021)	√							Cooja/Contiki	Residual Energy Link Quality and ETX		√		Improvements on energy consumption and mobility consideration.
(Abubakar Gidado Halilu, Martin Hope, Haifa Takruri, & Umar Aliyu, 2020)			√					Cooja/Contiki	Hop Count and ETX		√		Consideration for energy scavenging nodes
(Aliyu, Takruri, Hope, & Halilu, 2020)	√							NS3	OLSRv2				Mainly used for disaster network with improvements on energy consumption.
(Lamaazi et al., 2015)	√	√						Cooja/Contiki	Residual Energy and ETX		√		Mobility consideration with higher node densities but no practical improvement observed
(Shakya et al., 2017)	√							Cooja/Contiki	Residual Energy				For industrial applications with no mobility consideration
(Khan et al., 2016)	√			√				Cooja/Contiki	Network Optimization metric				Used multiple sink nodes to improve throughput but no mobility consideration, no practical experiment and incompatibility issues
(Banh et al., 2016)	√							Cooja/Contiki	Node energy and ETX		√		Achieves load balancing but no mobility consideration

(Riker et al., 2017)	√							Contiki	Battery model and routing aggregation	√			Network lifetime improvement but no mobility consideration
(Khelifi et al., 2015)	√							Cooja/Contiki	Energy Consumption and ETX		√		Network lifetime improvement but no mobility and practical implementation
(Gaddour et al., 2015)		√						Cooja/Contiki	Modified DIO message				improvements in stability and energy consumption but no support for mobility
(Gara et al., 2015)		√						Cooja/Contiki	Modified trickle timer algorithm				Improvements in packet delivery and mobility management schemes but used continuous timers that works like trickle creating more overhead
(Anand & Tahiliani, 2016)		√						Cooja/Contiki	Modified trickle timer algorithm				Creates more flexibility but objective function depends fully on RSSI
(Barcelo et al., 2015)		√						MATLAB	Mathematical models for blacklisting and routing				Improvement in packet delivery but increase energy consumption
(H. Kharrufa et al., 2017)		√						Cooja/Contiki	ETX and LQI		√		Improvements in delay, energy consumption and packet delivery but not enough mobility management schemes
(H. Kharrufa et al., 2018)		√						Cooja/Contiki	Game theory optimization (RSSI and LQI)		√		Improvements in delay, energy consumption, packet delivery and transmission change with respect to network conditions but no consideration to energy scavenging nodes.
(P.-O. Kamgueu et al., 2015)			√					Cooja/Contiki	QoS (ETX and Delay) and residual energy			√	Improvement on network lifetime and throughput but no mobility support
(Khelifi et al., 2015)			√					Cooja/Contiki	Packet loss threshold ($S_{threshold}$)				Improvement on network lifetime and throughput using combined cost metric but no mobility support
(Iwanicki, 2016)			√					TinyOS/TOSSIM	Link failure algorithm				Improvement on network reliability with suitable node collaboration but increased energy consumption

(Al-Kashoash, Hafeez, et al., 2017)				√				Cooja/Contiki	Traffic control algorithms (DCCC6, griping, deaf, fuse)				Improvement in energy consumption, delay and packet delivery but no mobility consideration
(Hellaoui & Koudil, 2015)				√				Cooja/Contiki	Standardised RPL OF				Improvement in packet delivery but only applicable to congested scenarios and no mobility consideration.
(Al-Kashoash, Amer, et al., 2017)				√				Cooja/Contiki	Buffer occupancy, queue delay and ETX		√		Utilizes adaptive transmission for maximization of throughput with Improvement in energy, throughput and delay but no mobility consideration
(Mayzaud et al., 2017)					√			Cooja/Contiki	Distribution detection algorithm				Prevents attack and locates attacker but more overhead and more deployment cost
(Zhu et al., 2017)						√		Not specified	Modified load balancing algorithm				Improvement in delay and load balancing but no mobility consideration
(Gozuacik & Oktug, 2015)						√		Cooja/Contiki	ETX, Link Quality and Latency		√		Utilized standardised RPL OF but no mobility consideration
(Di Marco et al., 2016)							√	Cooja/Contiki	MAC layer analytical model				Improved reliability and delay but no mobility consideration
(Tall et al., 2017)							√	Cooja/Contiki	Multichannel collaborative load balancing algorithm				Improvement in packet delivery and delay but no mobility consideration
(Lamaazi & Benamar, 2018)	√							Cooja/Contiki	ETX, Delay and Remaining Energy			√	Shows improvement in PDR and delay with equal energy consumption, but compared only with MRHOF and no mobility consideration
(Agarkhed & Kadrolli, 2017)	√							Not Specified	Remaining energy, crowdedness, and distance			√	Improves lifetime while reducing energy consumption, but causes more overhead
(Shakya et al., 2017)	√							Cooja/Contiki	ETX and energy consumption			√	Improves lifetime and indicates equal energy consumption, but compared only with MRHOF and no mobility consideration

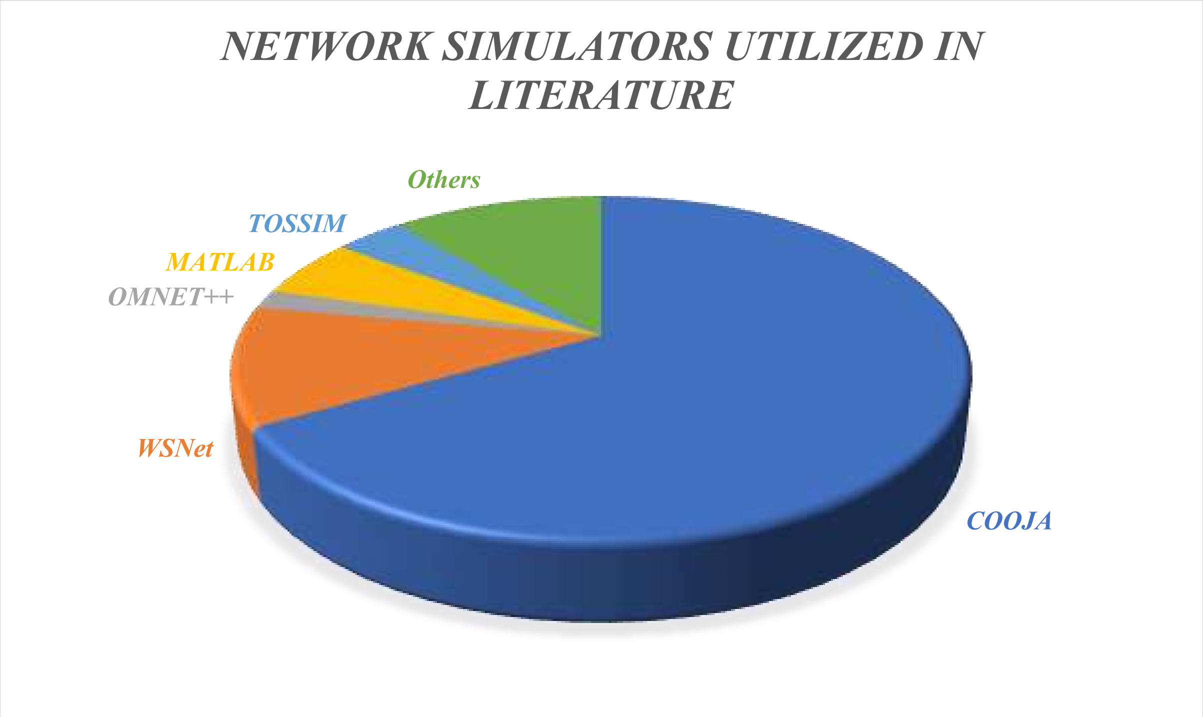


Figure 3- 6: Pictorial Representation of Related Works using different simulators in Table 3-1.

3.11 Summary

Routing in LLNs is still a very challenging aspect particularly as regards to ensuring load balancing, energy saving, and mobility. The native RPL OF which uses a single metric for routing decisions tend to have so many bottlenecks as effective routing performances cannot be guaranteed for different routing situations or when building a DODAG. In an unbalanced traffic, preferred parents having overloaded nodes consumes more energy than other member nodes which can result to the parent's energy to run down and eventually disconnecting from the network, none of the standardised OFs have any provision for load balancing. Consequently, this drawback affects network lifetime, reliability, and availability, making research on energy saving for load balancing a critical concern for routing in LLNs. Similarly, the standardised RPL OFs does not provide a suitable environment for mobility routing in LLNs, as the metric consideration used are hop-count and ETX thereby neglecting the effects of end-to-end delay, link quality and received signal strength. Routing accuracy for mobility in building a DODAG in RPL will be best determined using metrics that are mobility inclined defined in the modified OF.

Chapter Four

Enhanced Mobility Support Routing in IoT

4.1 Introduction

Mobility in network applications is one of the many issues affecting effective network performance and most research works on mobility in IoT particularly on RPL focus on multi-hop communication between nodes as well as the relationship between trickle timer algorithms and objective functions. This research proposed an enhanced mobility support routing protocol in RPL by combining three metrics in the objective function called *rpl_mobilityof.*

Within the chapter, this research implemented the *rpl_mobilityof* which is the OF for the enhanced mobility support routing where a simulation was conducted for different node densities and mobility models and was compared to the standardised RPL OF as defined in the RFC6550. Similarly, an analysis of the OF was made and a mathematical model will be developed. The chapter is organised as thus: Section 4.2 gives the proposed protocol design description and section 4.3 explains the proposed objective function for mobility outlining the various metrics utilized in the work. Section 4.4 explains the importance and role of the trickle timer algorithm; sections 4.5 and 4.6 explained the RSSI and Latency. Section 4.7 presents the protocol implementation algorithms. Section 4.8 describes in detail the simulation setup, results, and analysis with respect to some performance metrics. Lastly, section 4.9 summarizes *rpl_mobilityof* performance.

4.2 Proposed Design Description

Internet of things popularly known as IoT encompasses different devices or things to connect, communicate, collaborate, and carry out decisions from remote locations. It is an emerging area of research that is highly considered in both industry and academia. Due to the rapid evolution of smart and mobile devices, machine to machine (M2M) communication has led to the diversion of research attention to IoT enabling technologies. It is believed that IoT is presently in its prime thereby making industries invest billions of dollars in the research for IoT and its enabling technologies while also making it a research focus in academia with many expected interests. Applications participating in IoT network have different requirements, topologies, and mobility considerations, but the proposed mobility routing protocol can suit most mobility needs in IoT. This research will present mobility scenarios like the ones in agriculture, healthcare, and military applications that will require forming the DODAG and RPL Instance as well as routing information to the LBRs (root) in a multi-hop manner. The design does not involve modifying the RPL trickle timer, but rather only the objective function will be modified and the intervals of sending DIO messages to effectively improve the performance of the mobility routing protocol.

4.3 Proposed Objective Function for Mobility

The proposed mobility objective function *rpl_mobilityof* utilizes the design concept of the default objective function available in COOJA simulator software, which is embedded in the Contiki Operating System but having additional metric consideration for path and preferred parent selection from leaf nodes to the DODAG root in an RPL Instance. These metrics include the RSSI which is based on the received signal strength indicator, Latency which represents the end-to-end delay in packet delivery and finally the hop-count which is based on the number of hops from a leaf node to the root. The default objective function uses a benchmark of one metric to swap between paths and preferred parents, but in this concept at least two metrics will be considered as default thresholds for choosing or changing path and preferred parent. This is done simply to reduce the frequency of handovers causing unnecessary overhead and eventually to improve network performance. The proposed routing metrics all work together within the objective function to enhance the path and preferred parent selection, making the objective function a requirement for mobile IoT applications. Changes in the value of RSSI indicates whether a node is static or mobile, for static nodes the RSSI transmissions do not change while for mobile nodes the RSSI transmission changes each time a node moves from its original position, the value of RSSI will be used as one of the thresholds for changing between different objective functions.

4.4 Trickle Timer Algorithm

The trickle algorithm continuously adjusts transmission window to ensure updated information are always circulated to the nodes, and its communication rate tends to increase logarithmically with respect to the nodes density using meek suppression mechanism with point selection. Using the density-aware mechanism coupled with a consistency model it provides a transmission guide to nodes, for instance, when nodes data are consistent with one another, the control messages are sent intermittently with just very few packets per hour, but when nodes data are inconsistent with its neighbours, the trickle algorithm quickly sends control messages in milliseconds to resolve the issue. It continuously tries to limit the frequency of control messages in the network to ensure reduced overhead thereby exponentially increasing the trickle algorithm timer for unchanging network.

The original trickle timer has not been modified; hence all simulations were done using the original version of the trickle timer algorithm as the focus of modification is the objective function and DIO message intervals. Trickle timer performs normally, as the control mechanism for adjusting data and control packets are done at the objective function level. Nodes update their information during communication exchange of DIOs or packets, that way when a node receives a message and realizes that message is inconsistent with its state, if realizes that its information is out-of-date and then it updates its information otherwise it simply forwards such message to the next hop. The detailed operation of the trickle timer algorithm was discussed at section 2.5.1.3 of this thesis.

Algorithm 4- 1: Trickle Timer Algorithm

```
Algorithm 4.1: Trickle Timer
Begin:
Initialise trickle timer
        I = Imin
        Start new interval (loop)
                I ← I X 2
                C←0
        If (Imax ≤ I) then
                I ← Imax
                End if
                t ← random [I/2, I]
        If (Received consistent transmission)
                C ← C + 1
        If (Received inconsistent transmission)
                I ← Imin
        Random timer expires
        If C < K, then // K is a constant defined by the user to specify interval, 1-5seconds
                Transmit scheduled DIO
        else
                Suppress scheduled DIO
        End if
End
```

(H. D. Y. Kharrufa, 2018) used a reverse-trickle algorithm that is based on RSSI readings, when a node receives a packet it checks for the RSSI value of the packet and compare it to the last received RSSI value. If the latest reading of the RSSI value is lower than a defined constant (K_{RSSI}), then the reverse trickle timer algorithm sets in otherwise the native RPL trickle timer continues. The idea of the reverse trickle timer claims that moving nodes do not always change parents thereby leaving the preferred parent selection switching to objective function. This is the main reason why the approach focusses mainly on modifying the objective function than the trickle algorithm to reduce the number of overheads caused by computational complexities of using different trickle timer algorithms.

Additionally, according to (Levis & Clausen, 2011), the trickle algorithm has been designed to be highly robust in adapting to different networking environments coupled with several simple and tightly integrated mechanisms. It is highly efficient and any attempt to reduce the number of transmissions or responses can result to failures in edge cases thereby making the efforts not worth the risk. Therefore, the trickle timer algorithm is not perfect but changing any trickle timer algorithm will require detailed experimental evidence, in that such modification does not affect algorithm edge cases or assumptions.

Furthermore, tweaking trickle timer algorithms in dynamic and dense network environments particularly in wireless mobile sensor networks creates a lot of coordination issues and coordination is required to determine the best set of transmissions, making the need for its improvement to be weighed against generated computational complexities.

4.5 Latency

Latency is another routing metric used in this research, although it is dependent on the speed of the transmission rate utilized and other inherent delays incurred from devices along the transmission route. It simply means the amount of time required for a packet to traverse the network from sender to receiver, sometimes from sender to receiver and back. It works hand in hand with throughput, while latency specifies the amount of time needed for a particular action to be carried out and completed, throughput measures the overall number of such actions within a given time. Normally, continuous flow of data presents very little propagation delay on throughput as compared to messages awaiting acknowledgement from sent packets which could result in increased delay while reducing network throughput. Latency or end-to-end delay is a vital aspect in networking particularly for real time applications requiring acknowledgement. However, UDP transmission were used for simulations within the thesis to transmit control messages and to establish ranks as well as routing paths.

$$Latency = \sum \frac{Receive_{time} - Send_{time}}{Number\ of\ Hops\ (along\ route)} \qquad \mathbf{4.6}$$

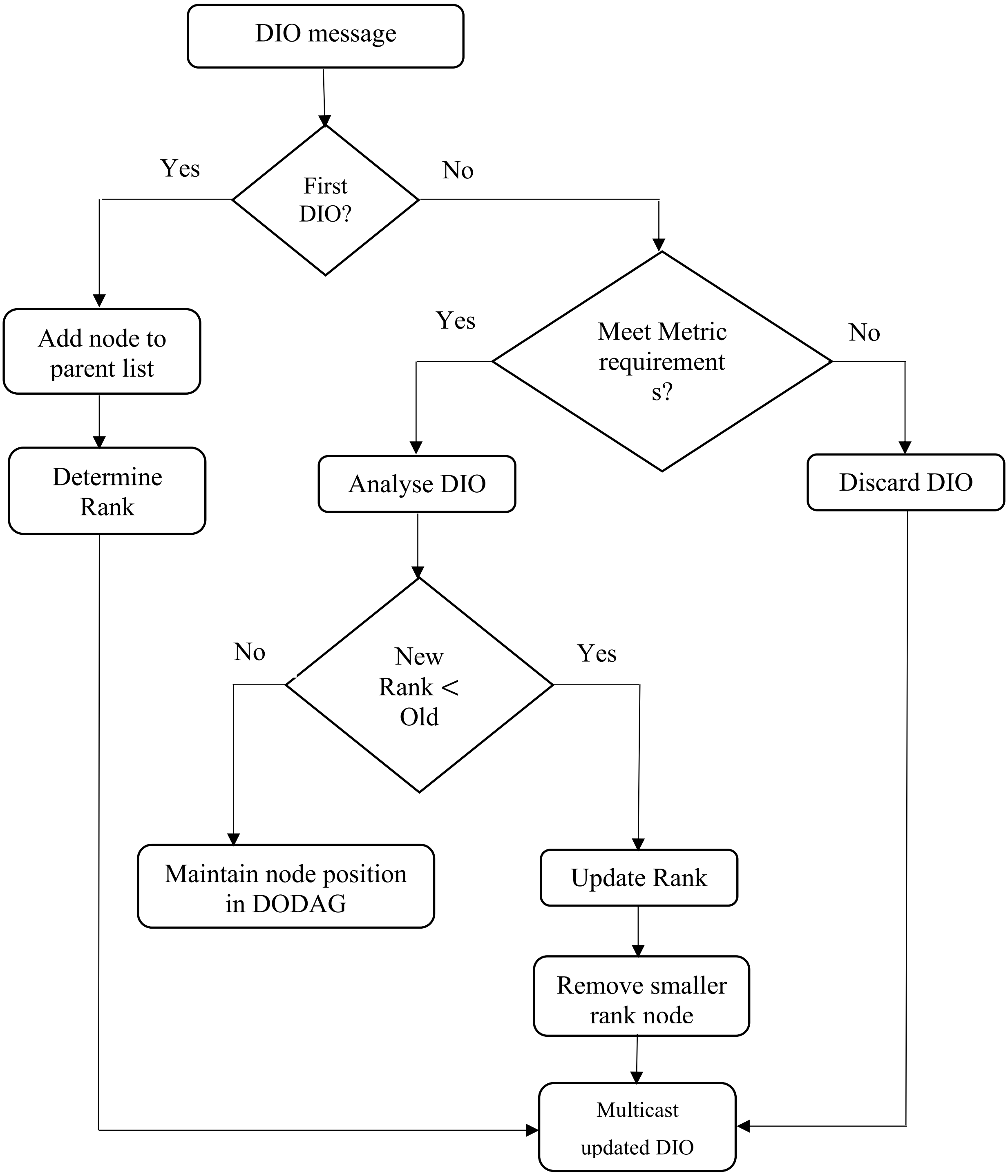


Figure 4- 1: DIO Message Flow

Figure 4-1 represents the flow of DIO messages in the DODAG once the simulation has been initiated. Latency is often described as the average time for communication that is transmitting packets from source to destination, and it can be mathematically represented as equation 4.6. Where both the send and receive times are extracted by an introduced modified function from the proposal. COOJA simulator has an inbuilt module which records latency on the *DIO_next_delay*, although packets in a network will have different delay times due to various reasons which could be since packets arrive their destination independently and at times packet queues causing irregular pacing. These inconsistencies, delay-time variations and irregular arrival of packets are known as jitter, mostly experienced in real-time communications which often leads to packet loss.

Although endpoints are responsible for ensuring that packets received are accurate and timely, however, in cases where such are not possible, it tries to attempt other means of correcting transmission and failure in that aspect which leads to packet loss. Similarly, network congestions are often caused because of nodes not been able to send traffic that is proportional to that which they receive, thereby overwhelming their buffers leading to packet drop. An issue on the endpoint preventing packets from arriving timely also causing the endpoint buffer to fill up making packets to continuously delay their arrival at the endpoint is another form of jitter called incipient congestion. The presence of incipient congestion signifies rapidly occurring jitter.

4.6 Protocol Implementation

As was clearly mentioned in previous sections, RPL is initially developed for static networks, but in other to implement mobile scenarios a mobility plugin must be downloaded in COOJA and incorporated with Bonnmotion for the implementation of different mobile scenarios. The mobility extension file format in COOJA is *".dat"* which was used to generate different mobility scenarios for different models, enabling the advantages of testing and evaluating various solutions while effectively comparing with other solutions. When a mobility scenario was compiled using the Bonnmotion tool, the resultant output was converted by an application in Bonnmotion known as *WiseML*, which was then finally converted to a *"position.dat"* file for the different scenarios which is fully supported by the mobility plugin in COOJA. The algorithm for the implementation of the mobility OF is given at Algorithm 4-2.

While compiling the mobility scenario in Bonnmotion, specifications were clearly stated for the following points; the mobility model used (Random WayPoint and Random Walk), the number of nodes used (10, 20 and 30), overall running time for simulation, start time for simulation, boundaries for mobile nodes, maximum and minimum speeds for nodes in the simulation as well as the height of node movement (applicable in 3D scenarios). Algorithm 1 explains the different procedures needed to achieve enhanced mobility routing protocol in IoT.

Algorithm 4- 2: Enhanced Mobility Support Routing Algorithm

```
Algorithm
Begin:
Initialise Algorithm
// set mobility counter function
hop_count_mobility = 0;
Void SetHopCountForMobility (hops)
        {
        hop_count_mobility = hops;
        }
// get RSSI
rssi ← packetbuf_attr (PACKETBUF_ATTR_RSSI); // indicates node displacement
// get latency
latency ← p → dag → instance → DIO_next_delay; // gives packet delay

Input: Received DIO
        For every Received ← DIO do
                If (rssi == 0) then
                {
                        hop_count_mobility = 0;
                        latency = 0;
                }
        /* Do not penalize the ETX when collision or transmission error occurs. */
                        If (status == MAC_TX_OK | | status == MAC_TX_NOACK) then
                        {
                        // calculate the hop metric
                        hop_metric = hop_count_mobility * RPL_DAG_MC_DIVISOR;
                        // calculate the new metric
                        new_metric = combination of (rssi, latency, hop_metric);
                        nbr → link_metric = new_metric;
                        }
                        Return link_metric;
                        Endif
                Endif
        Multicast the updated Link_metric
End
```

It starts off by initializing hop counts and obtaining *RSSI* values from the packet buffer attribute function, once the *RSSI* value changes from the initial value it automatically signifies node displacement hence the mobility OF is initialised.

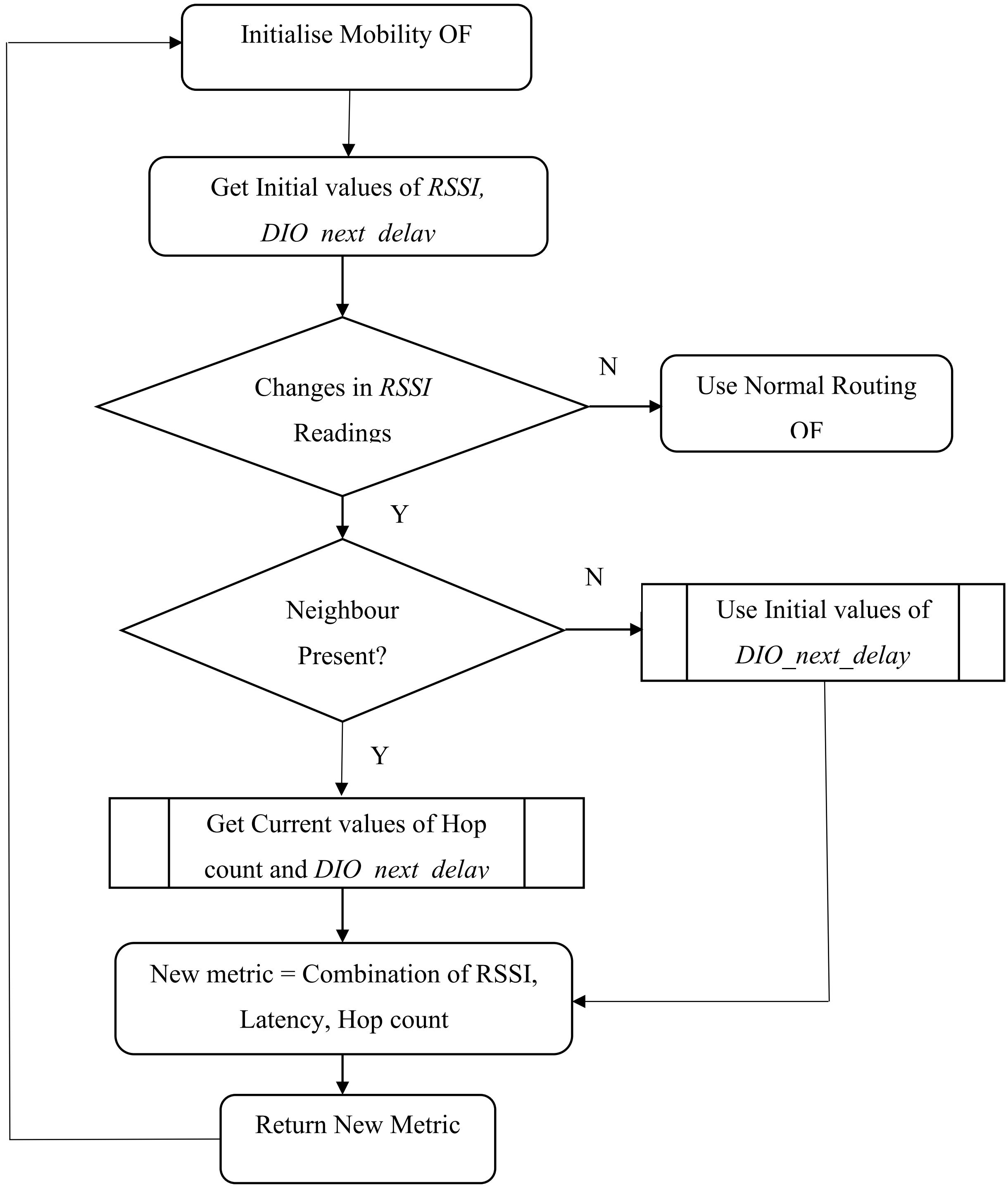


Figure 4- 2: Process Flow for Mobility OF

Thereafter, the process is run continuously, and a final link metric value is returned at every iteration. Figure 4-2 describes a clearer process flow for mobility OF highlighting the different steps required for the iteration. The computation of these three metrics uses a linear combination mode to combine the different values in a single readable value for decision making (see appendix 7).

Random Walk (RWK): it is one of the most widely utilized individual mobility models used in wireless networking performance evaluation in representing individual node movements, having the inherent capability for sudden change in speed and direction of nodes. It is sometimes referred to as the *Brownian motion*, the nodes in this model moves randomly with different speed and direction in memoryless (not storing speed or path) manner making it susceptible to sudden change that will result to unrealistic movement of nodes. When a node reaches the predefined boundary, it bounces with a different angle specified by the incoming direction. It was originally designed to represent erratic movement in an unpredictable medium in physics (Bai & Helmy, 2004).

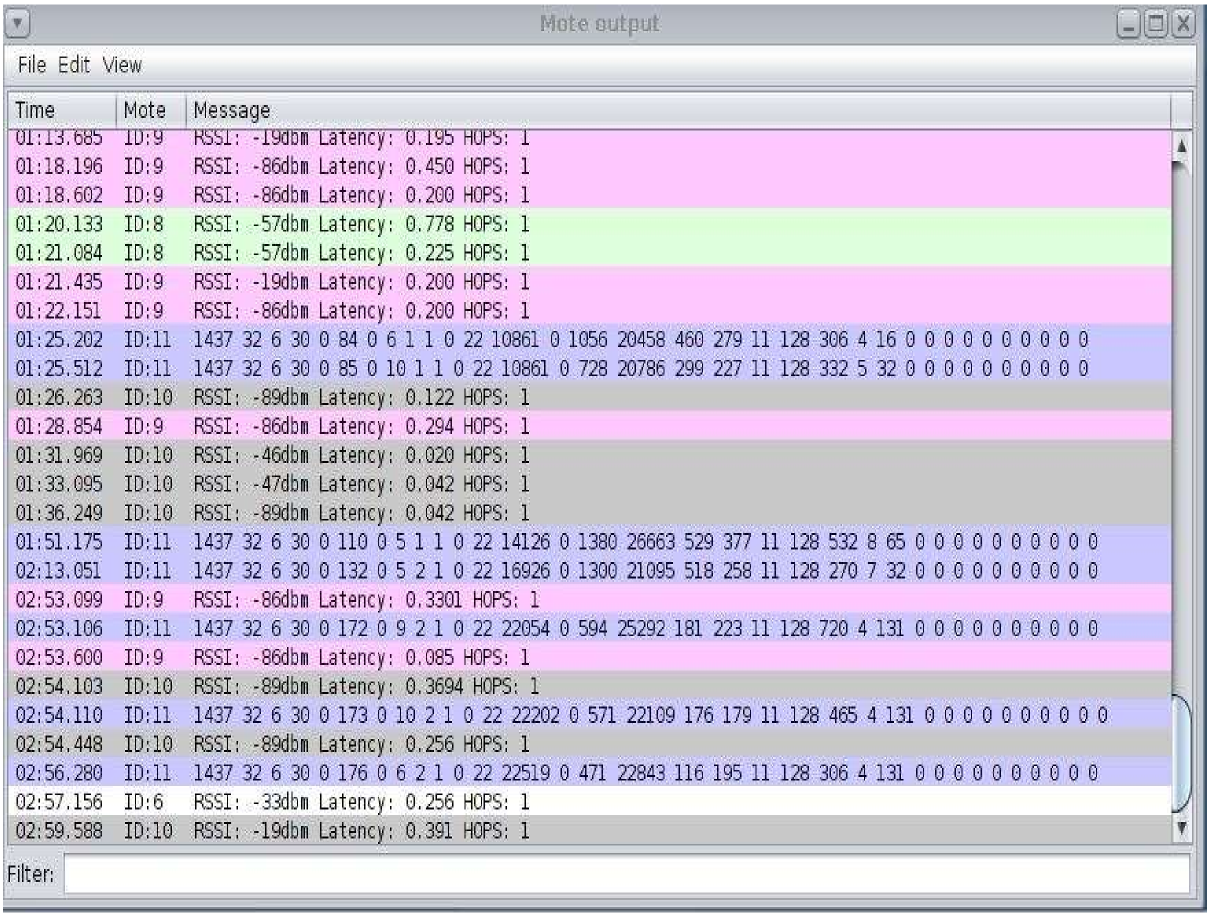
Random Waypoint (RWP): this is also another model utilized in this research, it is mostly used in ad hoc network simulation due its simplistic implementation on Network Simulator (NS2 and NS3) and it is an extension of Random Walk. It considers pause times between any change in speed and direction (Musolesi, Hailes, & Mascolo, 2004), it works in the following steps: by choosing a random destination, choosing a speed, move to the destination, wait for a specified pause time and repeat the process. This models behaviour is dependent on two important values the Max speed and pause time which works hand in hand, if pause time is long and max speed is low then the network is said to be stable, if the other way round then highly dynamic (Bai & Helmy, 2008).

4.7 Simulation Setup, Results and Analysis

4.7.1 Simulation Setup

The enhanced mobility support routing was implemented in COOJA simulator which is a wireless sensor network simulator embedded in Contiki Operating System 3.0. COOJA network simulator is an open-source software in Contiki OS that has all the detailed implementation files of RPL with all source codes available making RPL modification easy and straightforward. The two objective functions available in COOJA simulator are OF0 and

MRHOF which uses hop-count and ETX metrics respectively. RPL was originally designed for static networks, but a mobility plugin can be used to plot different movement patterns for mobile nodes, similarly, Bonnmotion (de Waal & Gerharz, 2003) tool was also used to depict different mobility scenarios using the various mobility models available with the software. Different researchers try to use different approaches to include mobility mostly by tweaking the trickle timer algorithm to include RSSI readings which could potentially cause additional overhead from conducting unnecessary handovers. Others try to use reverse trickle algorithm known as D-RPL (H. D. Y. Kharrufa, 2018) which is also initiated by changes in the RSSI readings to improve responsiveness in RPL, and it works in conjunction with the objective function as well as information obtained from neighbour nodes to make path and preferred parent selection.



```
01:13.685  ID:9   RSSI: -19dbm Latency: 0.195 HOPS: 1
01:18.196  ID:9   RSSI: -86dbm Latency: 0.450 HOPS: 1
01:18.602  ID:9   RSSI: -86dbm Latency: 0.200 HOPS: 1
01:20.133  ID:8   RSSI: -57dbm Latency: 0.778 HOPS: 1
01:21.084  ID:8   RSSI: -57dbm Latency: 0.225 HOPS: 1
01:21.435  ID:9   RSSI: -19dbm Latency: 0.200 HOPS: 1
01:22.151  ID:9   RSSI: -86dbm Latency: 0.200 HOPS: 1
01:25.202  ID:11  1437 32 6 30 0 84 0 6 1 1 0 22 10861 0 1056 20458 460 279 11 128 306 4 16 0 0 0 0 0 0 0 0 0 0
01:25.512  ID:11  1437 32 6 30 0 85 0 10 1 1 0 22 10861 0 728 20786 299 227 11 128 332 5 32 0 0 0 0 0 0 0 0 0 0
01:26.263  ID:10  RSSI: -89dbm Latency: 0.122 HOPS: 1
01:28.854  ID:9   RSSI: -86dbm Latency: 0.294 HOPS: 1
01:31.969  ID:10  RSSI: -46dbm Latency: 0.020 HOPS: 1
01:33.095  ID:10  RSSI: -47dbm Latency: 0.042 HOPS: 1
01:36.249  ID:10  RSSI: -89dbm Latency: 0.042 HOPS: 1
01:51.175  ID:11  1437 32 6 30 0 110 0 5 1 1 0 22 14126 0 1380 26663 529 377 11 128 532 8 65 0 0 0 0 0 0 0 0 0 0
02:13.051  ID:11  1437 32 6 30 0 132 0 5 2 1 0 22 16926 0 1300 21095 518 258 11 128 270 7 32 0 0 0 0 0 0 0 0 0 0
02:53.099  ID:9   RSSI: -86dbm Latency: 0.3301 HOPS: 1
02:53.106  ID:11  1437 32 6 30 0 172 0 9 2 1 0 22 22054 0 594 25292 181 223 11 128 720 4 131 0 0 0 0 0 0 0 0 0 0
02:53.600  ID:9   RSSI: -86dbm Latency: 0.085 HOPS: 1
02:54.103  ID:10  RSSI: -89dbm Latency: 0.3694 HOPS: 1
02:54.110  ID:11  1437 32 6 30 0 173 0 10 2 1 0 22 22202 0 571 22109 176 179 11 128 465 4 131 0 0 0 0 0 0 0 0 0 0
02:54.448  ID:10  RSSI: -89dbm Latency: 0.256 HOPS: 1
02:56.280  ID:11  1437 32 6 30 0 176 0 6 2 1 0 22 22519 0 471 22843 116 195 11 128 306 4 131 0 0 0 0 0 0 0 0 0 0
02:57.156  ID:6   RSSI: -33dbm Latency: 0.256 HOPS: 1
02:59.588  ID:10  RSSI: -19dbm Latency: 0.391 HOPS: 1
```

Figure 4- 3: Simulation Output for Mobility OF

Figure 4-3 shows the simulation output of the mobility OF, indicating the various performance metrics of different nodes in the simulation area. As indicated earlier, the mobility OF is based on the RSSI readings, the latency and hop count, the first reading displayed is the RSSI value which indicates whether a node has changed position from last received reading, if the position has changed then the node is considered mobile. The RSSI and Latency are the primary considerations for making routing decisions while the hop count is used as a tie breaker in very rear circumstances.

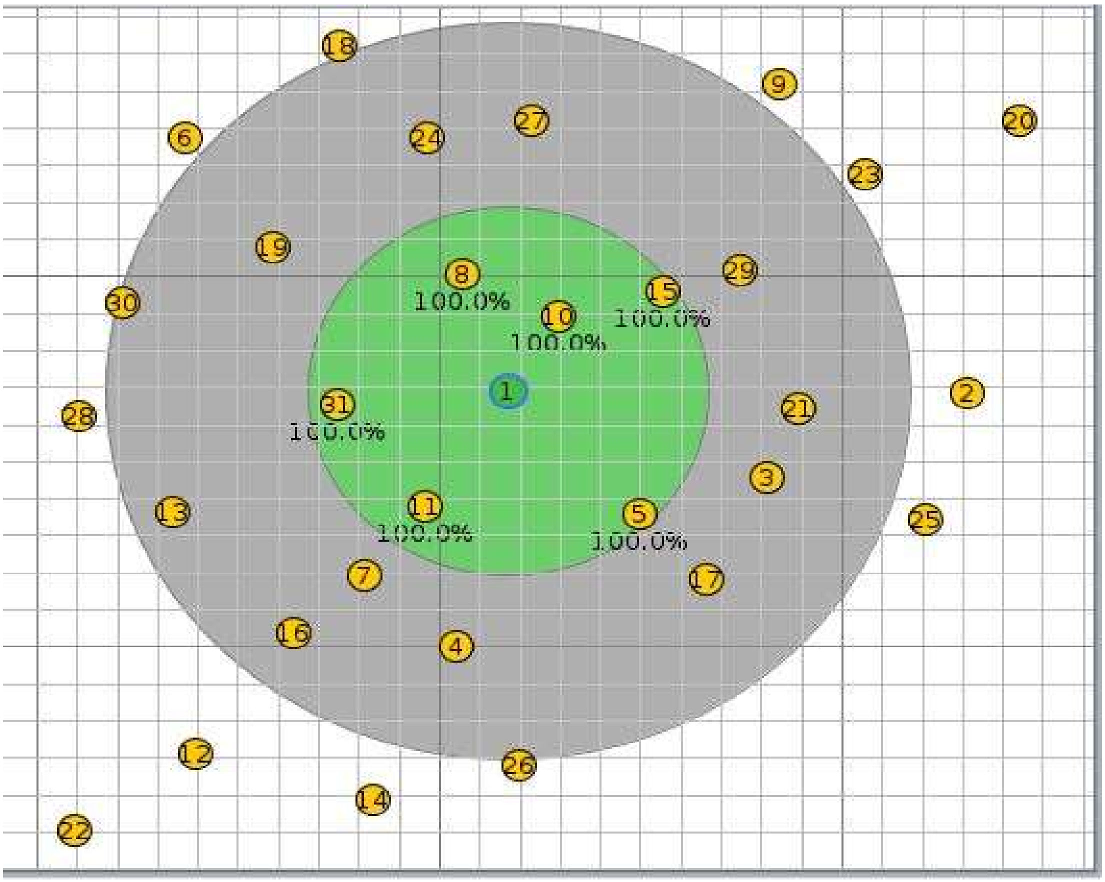


Figure 4- 4: Random Node Movement in COOJA simulator

D-RPL works with the assumption that all nodes in the network are mobile except the root node which is static, and this concept is not always the case with IoT devices. This research modelled a mobility OF in this work and all operations working on the background are displayed on the output window as in Figure 4-3. This work utilized different density of mobile nodes for simulation ranging from 10, 20 and 30 mobile nodes with one sink node within a 200m × 200m area as displayed in Figure 4-4. The green circle is the sink node, and the yellow randomly scattered nodes are sender mobile nodes.

This research utilized two mobility models the Random Way Point and Random Walk Mobility models with minimum speed of 0m/s for both models and a maximum speed of 4m/s and 2m/s respectively, the maximum and minimum pause time for both the two models are the same. The standard transmission and interference ranges were tested on the default mode of 50m each and a Bonnmotion tool was used to depict mobility scenarios and to build the mobility models for the simulation.

Table 4- 1: Simulation Parameters

Parameters Used	***Value Specification***
Network Simulator	*COOJA in Contiki OS 3.0*
Simulation Area	*200m × 200m*
Number of Nodes	*10,20,30 mobile nodes + 1 sink node*
Routing Protocol	***rpl_mobilityof***
Emulated Nodes	*Z1 Mote*
Transmission Range	*50m*
Simulation Time	*1800 seconds*
Node Deployment	*Random Positioning*
Radio Environment	*UDGM (Unit Disk Graph Medium)*
Radio	*CC2420*
Bonnmotion Tools	
X: Y area	*200m × 200m*
Minimum speed	*1m/s for all models*
Maximum speed	*4m/s and 2m/s respectively*
Minimum pause time	*4s*
Maximum pause time	*20s*
Mobility model	*Random Waypoint and Random Walk*
Interference range	*50m*

4.7.2 Simulation Results

The performance of the enhanced mobility objective function was tested against the default standardised objective functions available in COOJA RPL in terms of some performance metrics, these metrics include packet delivery rate (PDR), network convergence time, average energy consumption in nodes, end-to-end delay, and parent change. Each performance metric is important as they form the baseline for ascertaining the suitability of any modified routing protocol, PDR is the percentage of successfully delivered packets against the total number of sent packets. End-to-end delay represents the time period required for a member node to send packet to destination successfully, energy consumption represents the average amount of energy expended in transmitting a packet from source to sink within the stipulated transmission time, throughput measures the percentage of raw bandwidth utilized during transmission, overhead is the total number of DIOs, DAOs and DIS, scalability measures the ability of a routing protocol to meet the ever growing number of nodes in a given network and parent change indicates the number of times a node changes parent during the duration of the simulation.

The mobility OF does not perform well when there is no mobility because of the extra layer of processing required as compared to the standardised versions of the OF. Figures 4-5 and 4-6 shows the PDR of the standardised versions of RPL when implemented with mobility models. The approach shows better performance in PDR as compared to the standardised versions of the RPL protocols under mobility at the same transmission rate with very low number of dropped packets.

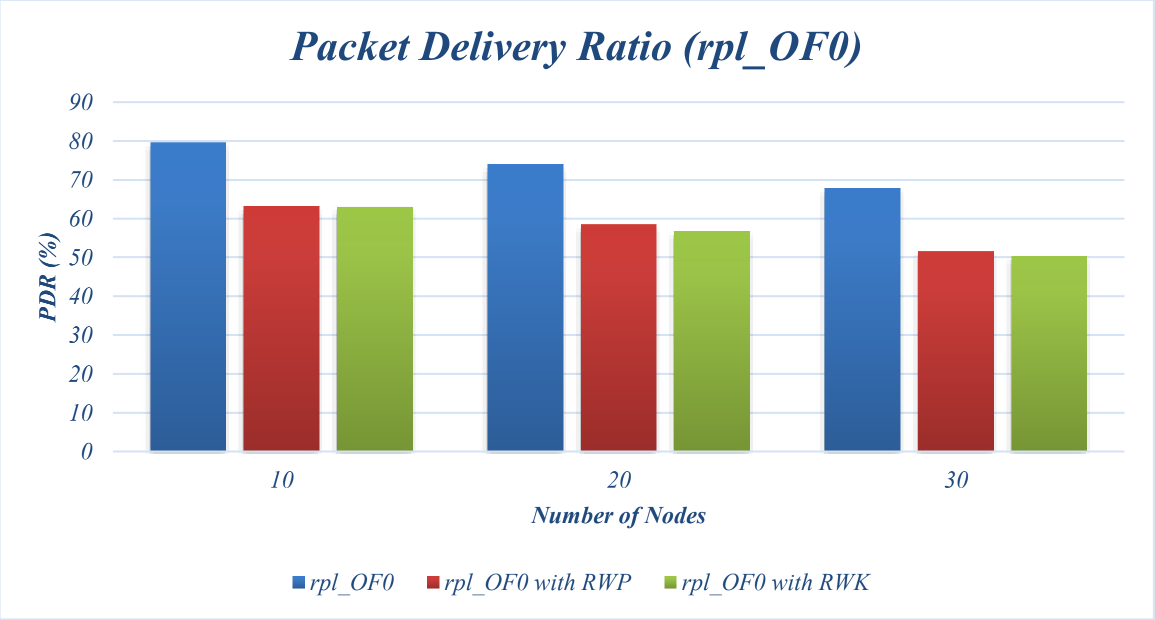


Figure 4- 5: PDR for *rpl_OF0*

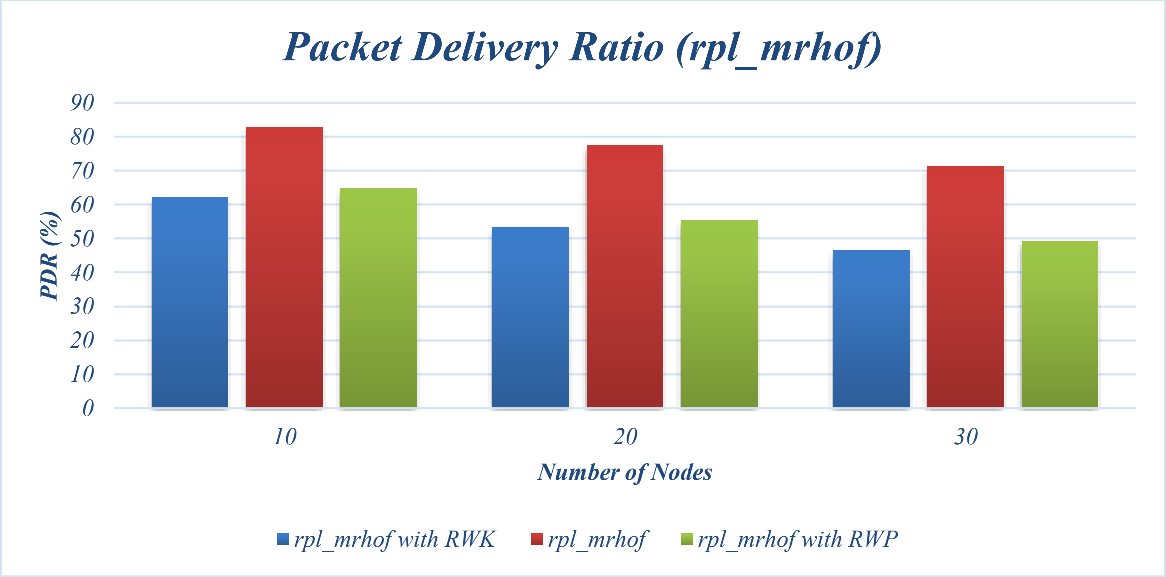


Figure 4-6: PDR for *rpl_mrhof*

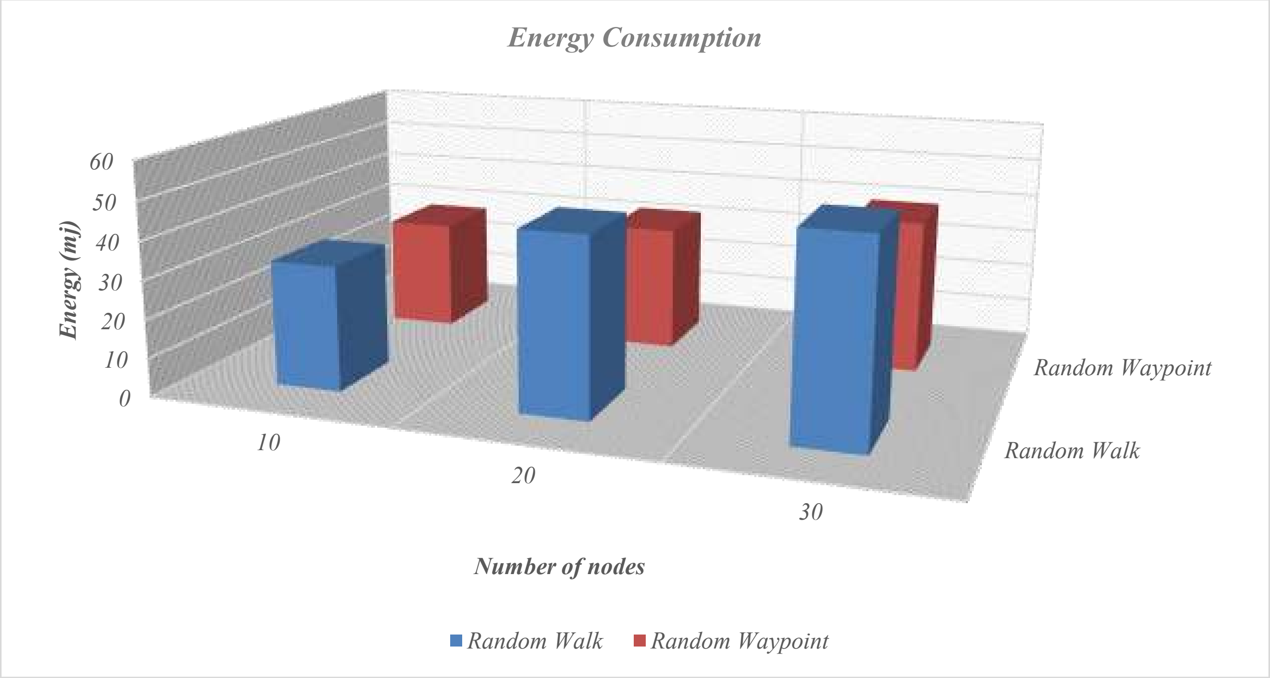


Figure 4- 7: Energy Consumption for *rpl_mobilityof*

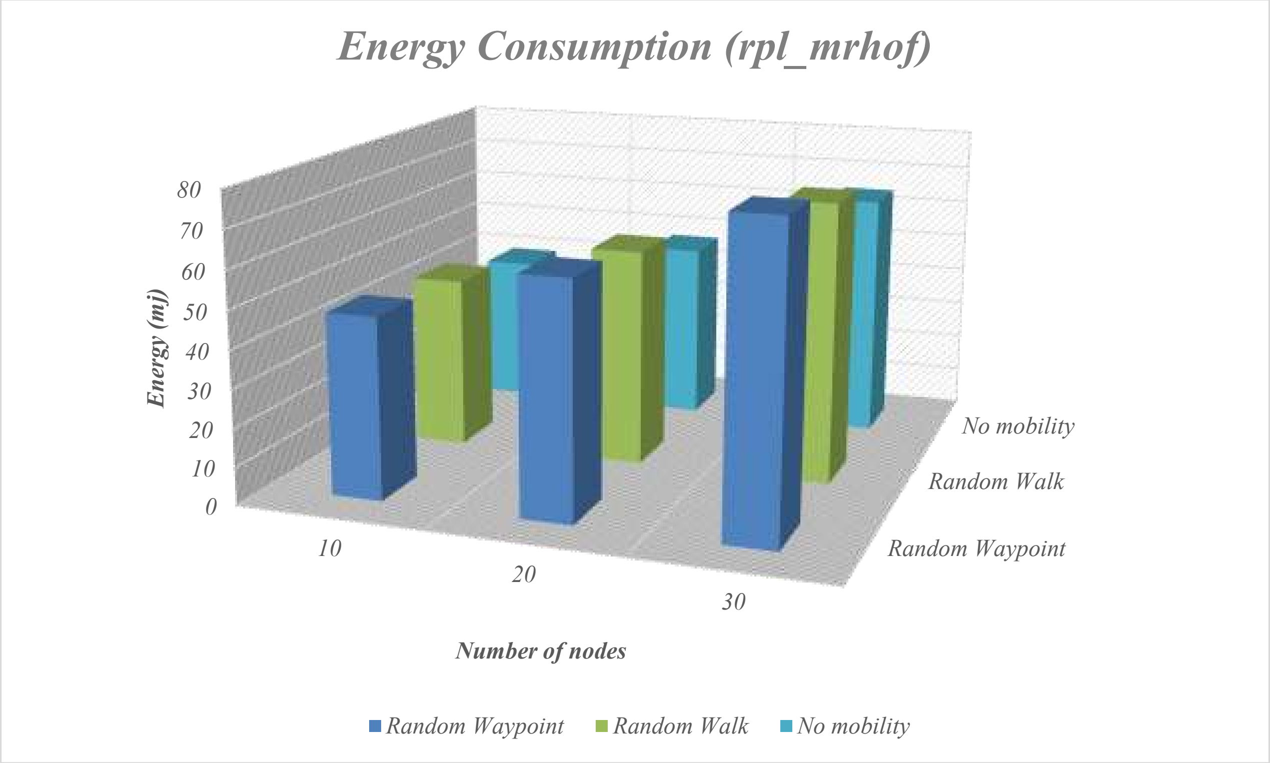


Figure 4-8: Energy Consumption for *rpl_mrhof*

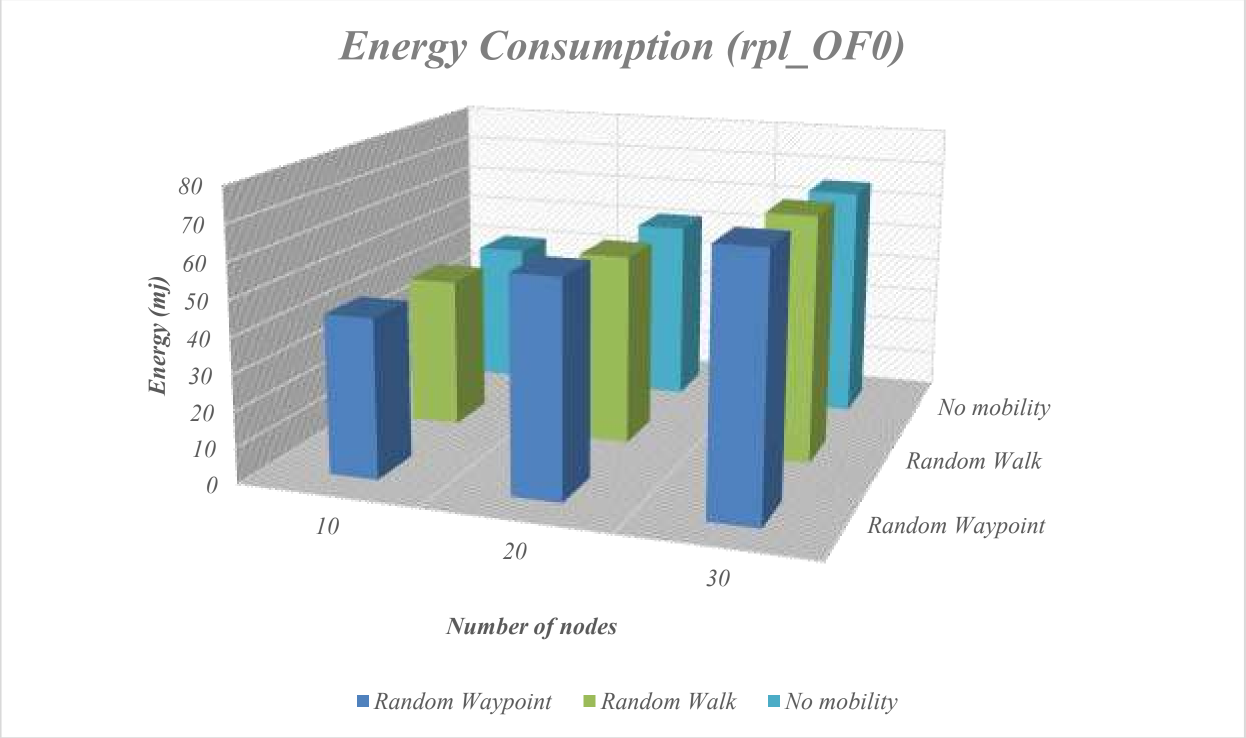


Figure 4-9: Energy Consumption for *rpl_OF0*

Figures 4-5 and 4-6 represents the PDR for the standardised RPL OFs for 10,20 and 30 nodes, *rpl_OF0* shows 80%, 74% and 68% of PDR implemented without any mobility model while 63%, 58% and 51% when implemented with Random Waypoint and 63%, 56% and 50% when implemented on Random Walk mobility models with the specified number of nodes respectively. Similarly, *rpl_mrhof* showed better performance in terms of PDR 83% when implemented without mobility model because it uses the estimated transmission count to determine the best route to forward packets. However, *rpl_OF0* performed better than *rpl_mrhof* in terms of PDR when implemented with mobility models, that is why this research used hop-count metric in combination with latency and RSSI to develop the mobility OF. Generally, *rpl_mobilityof* shows a PDR of 88% high in terms of mobility due to its adaptation for higher mobility situations, its optimization of the objective function to consider the RSSI readings and latency in combination with the hop-count allows it to make the most suitable routing decisions. The approach tends to reduce unwanted handovers which could cause packet loss while having a very active response to mobility. It reduces the dissemination of control messages by utilizing other metrics in the objective function for decision making. Comparing the results above with the default objective functions in RPL shows significant difference thereby justifying the proposal for developing a mobility aware objective function.

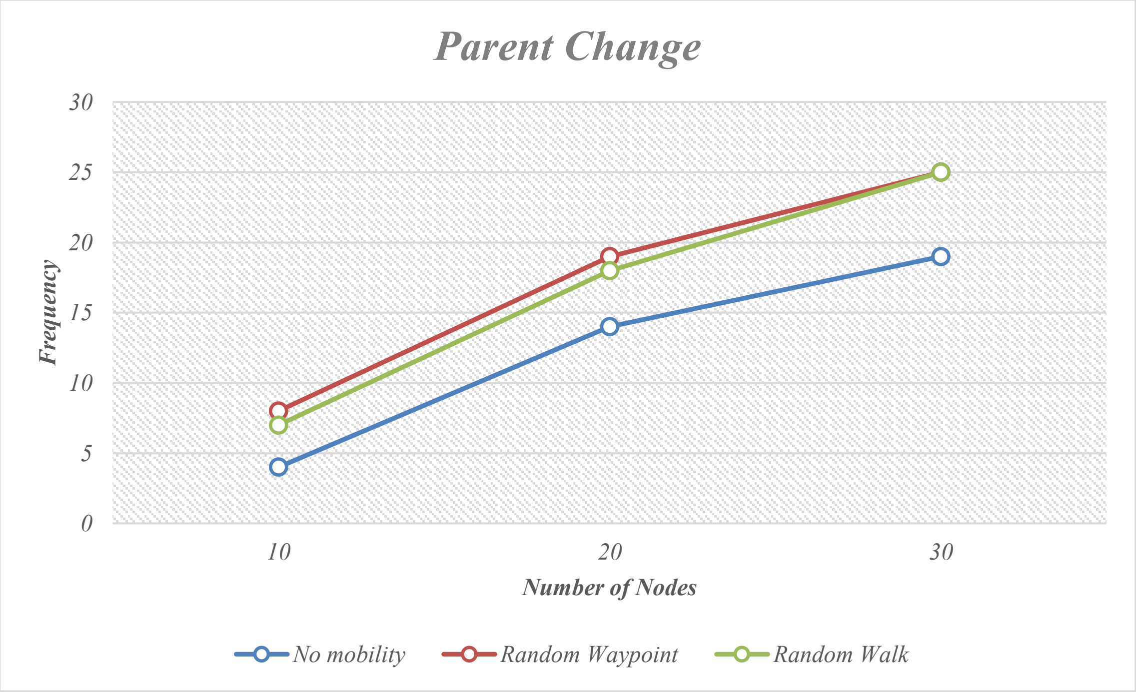


Figure 4- 10: Parent Change for *rpl_mobilityof*

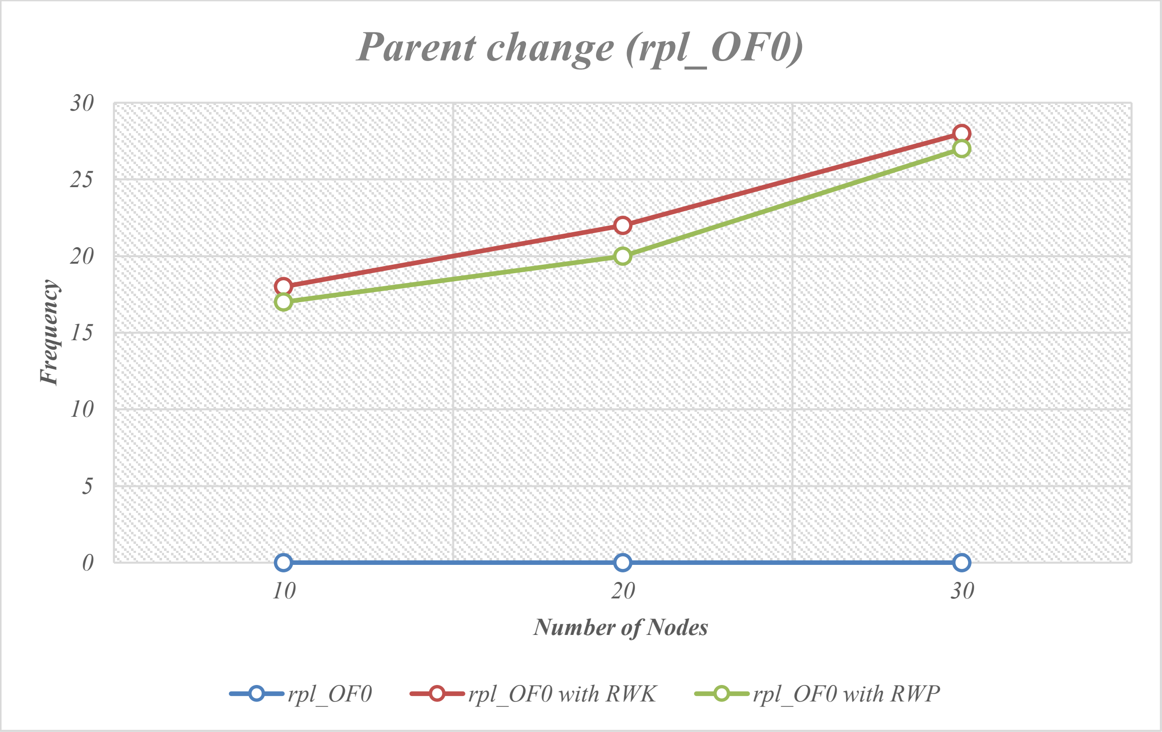


Figure 4- 11: Parent change for *rpl_OF0*

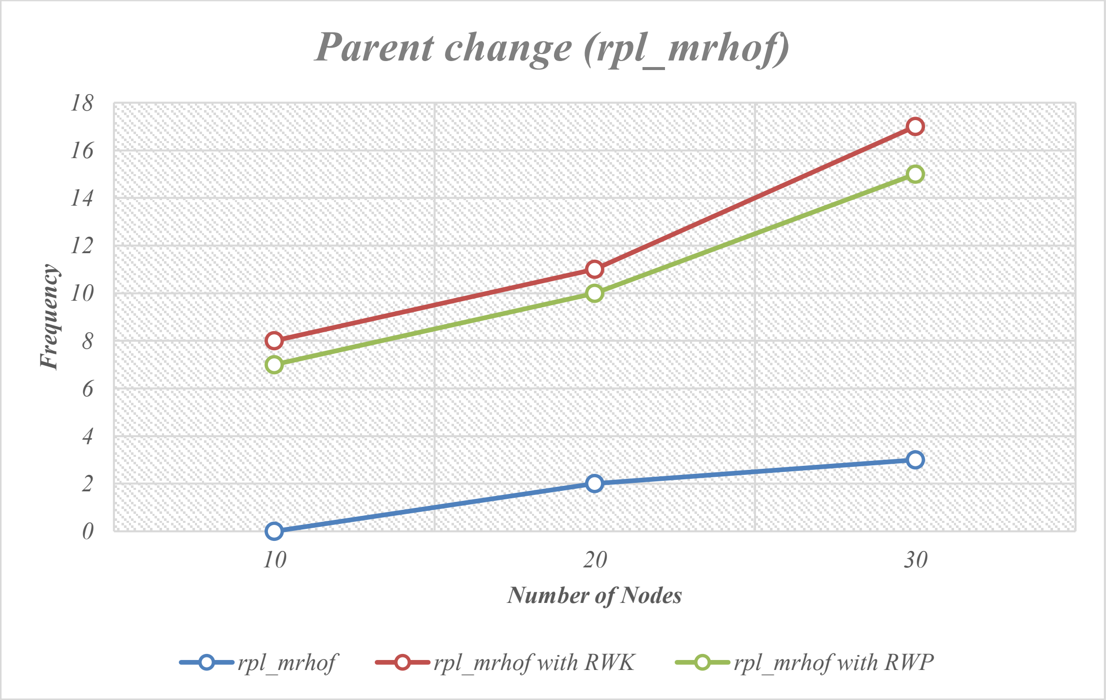


Figure 4- 12: Parent change for *rpl_mrhof*

Additionally, Figure 4-7 indicates the average energy consumption in the network for both the mobility models and the number of nodes, the results showed better energy consumption when *rpl_mobilityof* is implemented with mobility models than using the traditional *rpl_OF0* and *rpl_mrhof*, the difference between the energy consumptions is more than 38%. This is because the standardised objective functions were not originally designed to support mobility thereby having minimal response to mobility and that the protocol has a way of utilizing the changes in RSSI readings to trigger the mobility OF. Another factor explaining the difference is the fact that the protocol and the standardised protocols both use the same trickle timer algorithm which resets its sending of DIO messages upon noticing link failure or broken links.

Furthermore, the mobility objective function selects the most optimal value in terms of preferred parent and path selection minimizing the rate of retransmissions which leads to a lower energy consumption. *rpl_mobilityof* displays a reasonable level of mobility management from RSSI readings with the capacity of maintaining a fewer packet loss rate as compared to standardised OFs and the combination of different metrics for the OF makes it highly adaptive to mobile scenarios.

Another important feature for RPL routing is the path and preferred parent change, Figure 4-10 indicates the number of times a node switches parent while transmission is on-going. The approach incurs the highest number of parental changes as compared to the standardised versions, because the algorithm always determines the optimal path at every processing. This research changed the *PARENT_SWITCH_THRESHOLD* value from one to two meaning that two or more metrics will be used for path and preferred parent switch as opposed to one. But when mobility models were implemented on the standardised versions, they appear to switch paths and preferred parents regularly.

Similarly, Figure 4-13 displays the average end-to-end delay of the of the OF as compared to standardised versions, the OF shows better performance due to its optimal path and preferred parent selection. End-to-end delay analysis should be made in conjunction with packet delivery ratio, as it is concerned with packets that are delivered successfully while ignoring dropped or lost packets. Higher delay leads to packet loss and network congestion which overwhelms the capacity of buffer occupancy particularly in mobility situations, where different time stamps are applied on packets and verified upon receipt by LBRs (sink node). The proposal tends to show better performance all round in terms of parent change, energy consumption, packet delivery, and reasonable delay.

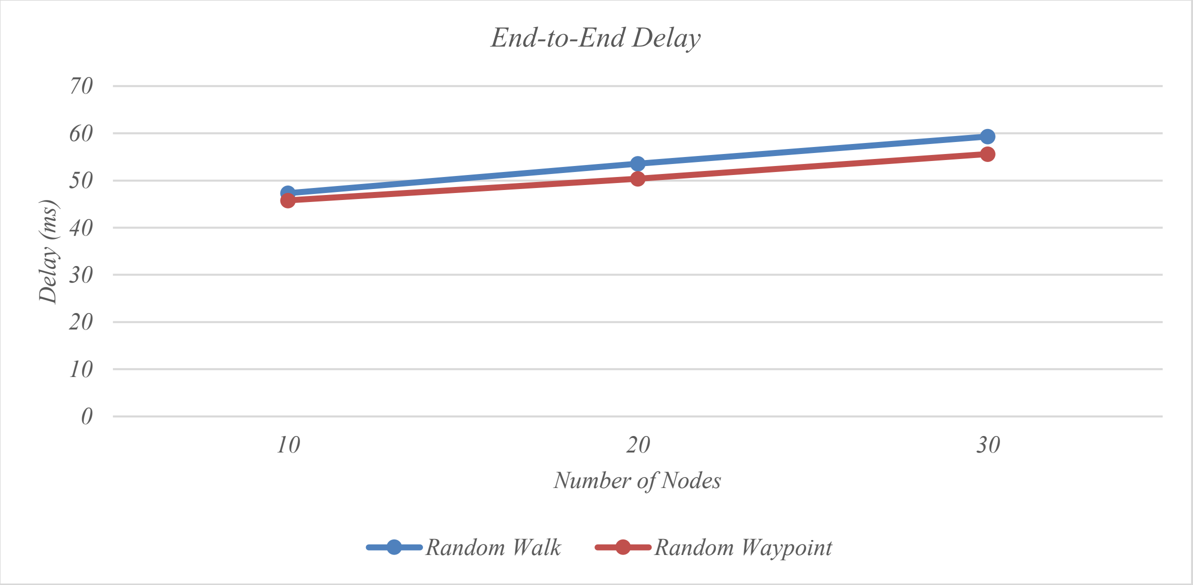


Figure 4- 13: End-to-end delay for *rpl_mobilityof*

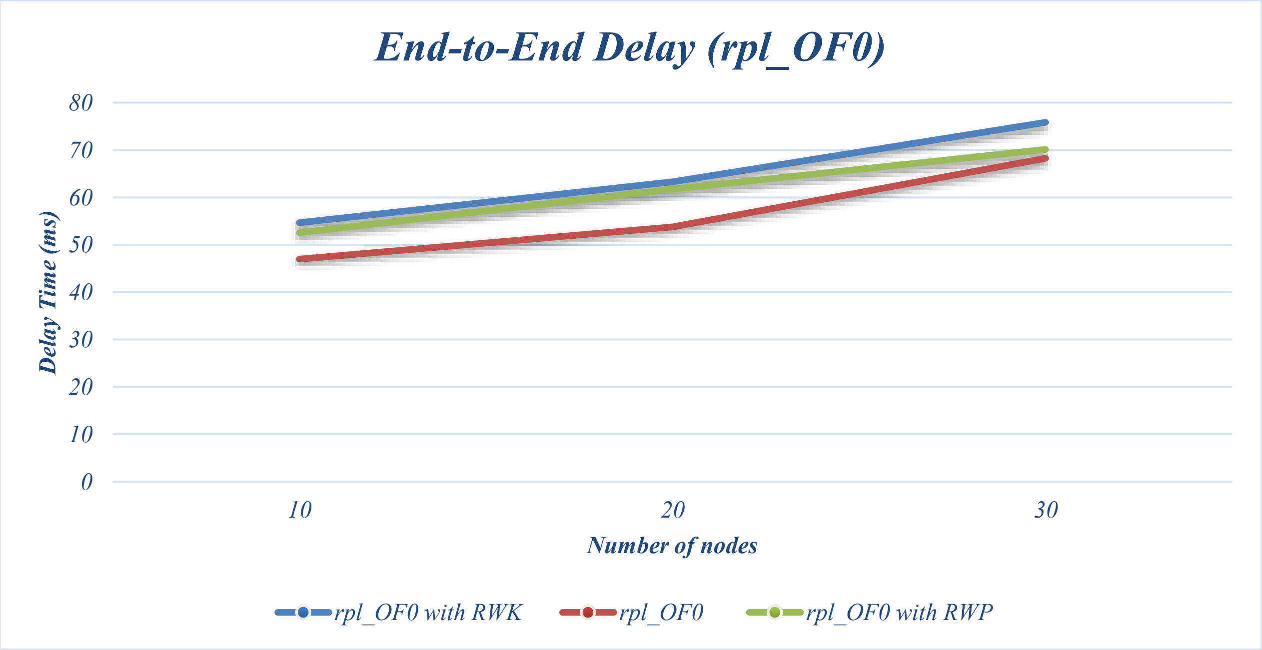


Figure 4- 54: End-to-end Delay for *rpl_OF0*

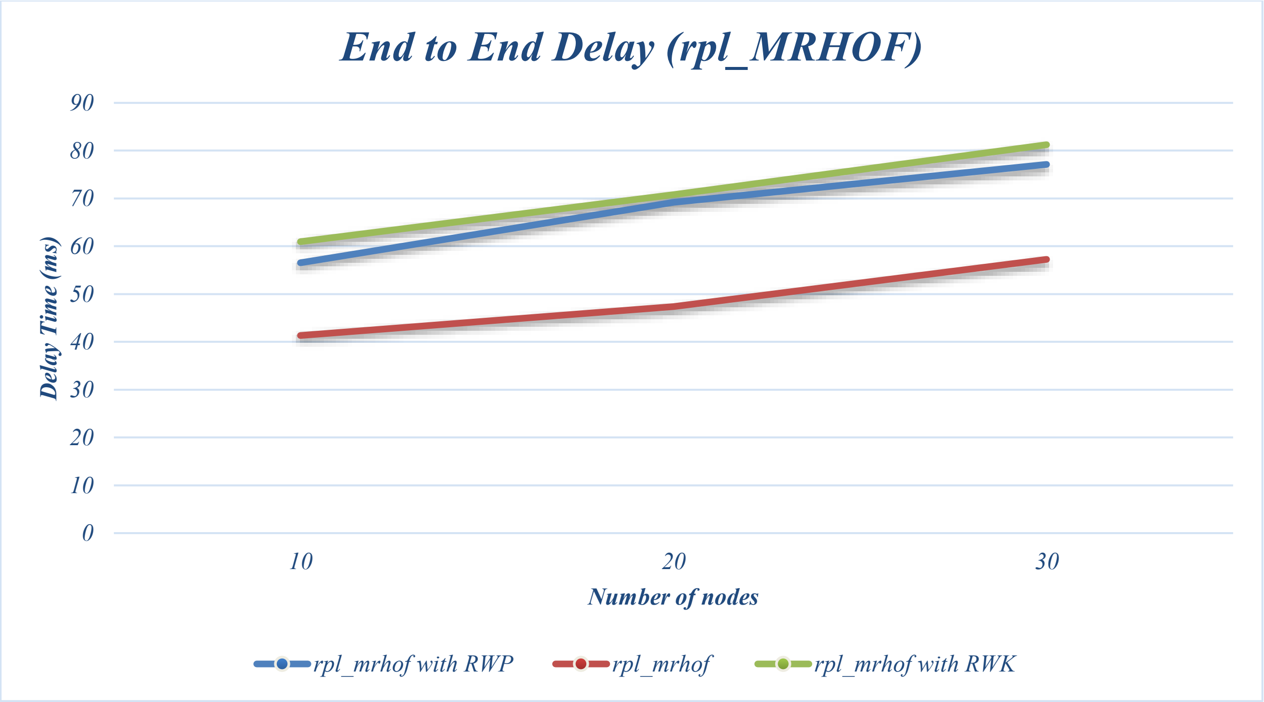


Figure 4-15: End-to-End Delay for *rpl_mrhof*

4.8 Summary

This chapter discussed the performance evaluation of *rpl_mobilityof* as compared with standardised versions of RPL OF for dynamic IoT applications in handling their mobility needs by providing highly robust, reliable, scalable, and efficient algorithms. Generally, *rpl_mobilityof* showed a better performance overall in terms of latency, packet delivery, energy consumption and parent change, which are implemented using two mobility models with varying node densities. The approach achieves an overall average of 88% PDR, 45.79ms latency and 28.90mj energy consumption as compared to the standardised version of RPL. Simulation results indicates how an OF can impact the entire performance of RPL protocol when handling mobile scenarios. The approach relies heavily on the changes of RSSI readings, which is a major threshold consideration in mobility routing, it is done in conjunction with other routing metrics such as latency and hop-count for path and preferred parent selection. The optimization of the three combined metrics in the objective function was crucial for improved routing performance in terms of mobility.

Chapter Five

Enhanced Energy Efficiency Routing in IoT

5.1 Introduction

Many IoT applications require energy efficiency as some of these applications are required to operate for long period of time, it is important for an energy efficient objective function to accommodate the different application requirements to ensure improved network lifetime. Energy efficiency is one of many issues affecting the network performance and most versions of energy enhancement in RPL focus on multi-hop communication between nodes as well as the relationship between trickle timer algorithms and objective functions. This work proposed an enhanced energy objective function called *rpl_energyof* to help improve energy balancing among nodes in the network.

In this chapter, the enhanced energy efficient routing protocol using different node densities and scenarios was proposed. This work implemented *rpl_energyof* and compared with the standardised versions of RPL OF and recorded certain areas of improvements. The remaining sections are organised as thus: Section 5.2 discusses the proposed protocol design description. Section 5.3 explains the proposed algorithm, section 5.4 explains the link quality indicator. Section 5.5 describes the simulation setup, results, and analysis with respect to certain performance metrics. Lastly, section 5.6 summarizes the chapter.

5.2 Proposed Protocol Design Description

The standardised design of RPL routing protocol in constrained devices utilizes the directed acyclic graph concept in terms of loop-free design, multipath, quick, and simple configuration, self-healing capabilities and minimized delay. The energy efficiency related to this protocol is considered as one of the weaknesses of the routing protocol that negatively impacts network reliability and lifetime as it was vividly discussed in the previous chapters. Additionally, RPL is mostly geared towards large scale distribution of both non-uniform and uniform low powered devices leading to irregular data traffic at the root node which causes unnecessary problems of energy consumption and load balancing. Nodes most affected with this problem are preferred parents that are one hop to the DODAG root node, and their failure could trigger a new retransmission as all other member nodes are connected to them. Limitation in preferred parent buffer can also lead to dropping of child nodes and congestion, as most overloaded connections of one hop neighbour to DODAG root are always fragile and energy consuming. Different network topologies present different issues of energy consumption particularly due to nodes taking multiple decisions for selection of preferred parent or direction of traffic flow in the network. RPL is by default a tree-based protocol, therefore load balancing and uniform energy distribution among nodes is challenging during a DODAG construction where DIO messages are exchanged.

5.3 Proposed Energy Saving Algorithm

The proposed algorithm of enhanced energy objective function *rpl_energyof* leverages a nodes lifetime by considering the remaining energy of a node when routing data traffic as well as determining the energy consumption percentage of a node. The objective function combines three routing metrics, the remaining energy of nodes which considers the energy of a preferred parent node at any point in time to make routing decisions, the ETX which is a default metric in *rpl_mrhof* and represents the number of transmissions to the root node, and the link quality metric that ensures reliable links are selected.

Algorithm 5- 1: Enhanced Energy Saving Algorithm

```
Algorithm
Input: Received DIO
Begin:
Initialise Algorithm
Lq_metric ← Link Quality Metric;
New_etx ← Calculated ETX value;
Energy_metric ← Total Remaining Energy of a node;
        For every Received ← DIO do
                If sender ∉ parent list, then
                        Update parent list ← sender;
                Endif
                If sender ∈ parent list, then
                        Calculate neighbour route metric;
                        New_metric ← combination of (New_etx,  Energy_metric, Lq_metric);
                        nbr → Link_metric = New_metric;
                Endif
                Return Link_metric
        Multicast the updated Link_metric
End
```

The amendment made on this proposed method is focused on the linear combination of these three-routing metrics which formed a new routing algorithm, hence neither the DIO message format nor the trickle timer algorithms were modified to prevent unnecessary routing complexities and overhead. A summary of the enhanced energy saving objective function is given in Algorithm 5-1.

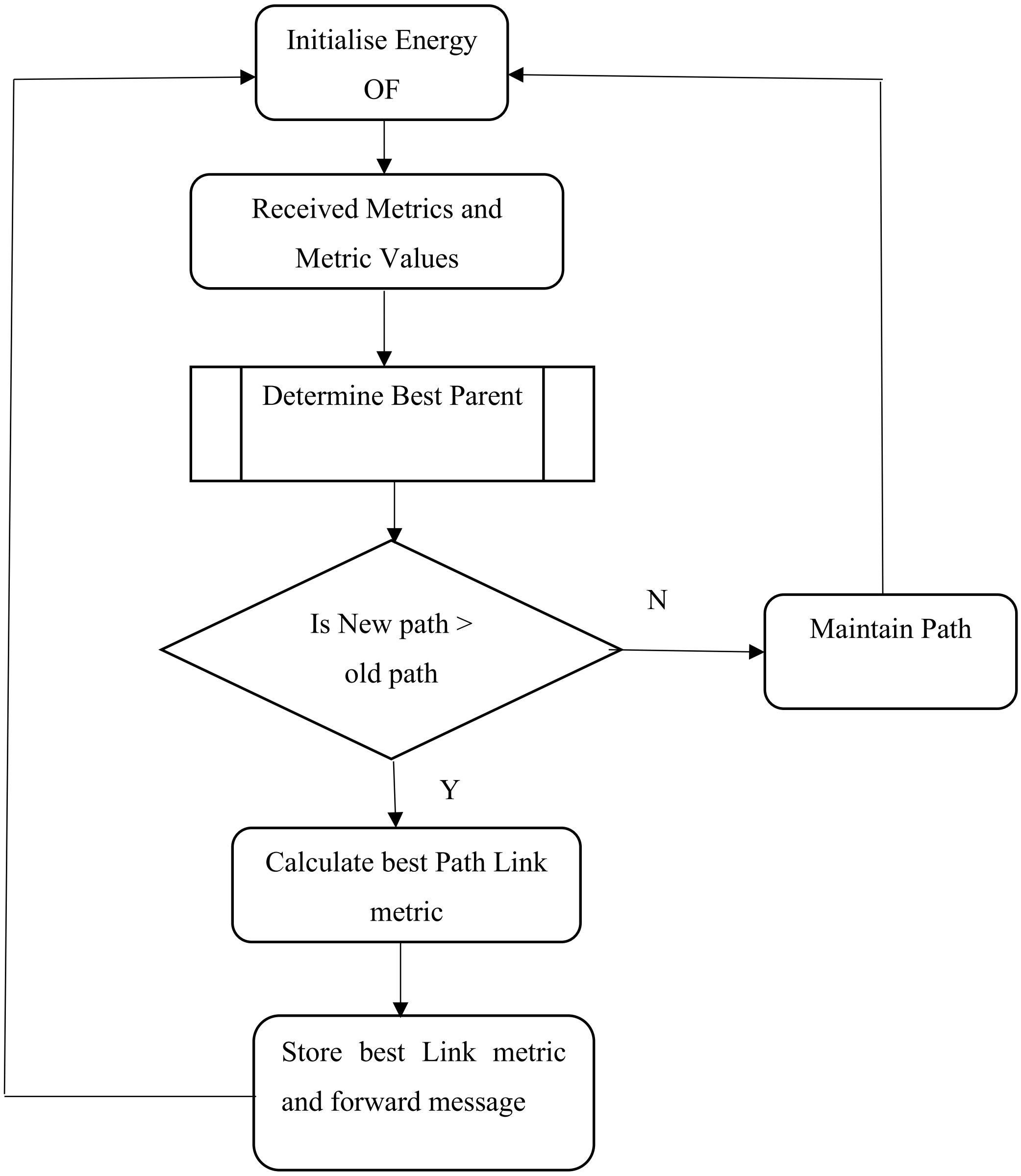


Figure 5- 1: Process Flow for Energy Efficiency Algorithm

Ideally, a DIO message header contains certain information about the DODAG that are useful in forming the route within the network, this information includes the version number, identifier number of the DODAG, the RPL InstanceID for the connection, the rank of preferred parent and the objective function used. The DIO in the proposed design has not been modified, but rather the message timings and objective function used in rank and route calculation. Irrespective of whether it is an upward route formation or downward route formation, when a DIO message is initiated for DODAG path construction in a network, neighbouring nodes receive the broadcasted DIO message and from the information contained in the DIO message it decides the preferred parent as well as the calculation of the rank. After these processes, the updated DIO message is broadcasted to neighbouring nodes and preferred parents will also receive the DIO message but will not act on it.

Furthermore, once a DODAG has been constructed successfully then the DAO messages are forwarded to know the suitable routes to the DODAG in both the upward and downward route setup. The proposed network setup has utilized the UDP transmission by aiming to remove the overhead caused by the TCP connection, and each preferred parent is aware of the energy metric and link metric values of their child nodes. Although the more child nodes a preferred parent has, the faster the energy consumption of the preferred parent, hence the need for different metric consideration for path calculation to ensure load balancing.

5.4 Proposed Design Description for Energy Scavenging Nodes

Most researchers focus on modifying the RPL protocol to accommodate mobile nodes or reduce energy consumption among nodes. IoT encompasses an extensive range of applications with varying standards and technological implementations with some having different network consideration, dissimilar node arrangements and different energy modes. *rpl_QoSof* is designed for networks whose nodes are attached to an energy source whether on mains or energy scavenging modes and combining the performance of both OF0 and MRHOF while eliminating their disadvantages. It will be tested on nodes that are static in nature and require regular or irregular energy source in a multi-hop environment. The design does not tamper with the trickle timer algorithm and the performance will be evaluated with the standardised versions of the objective functions with regards to packet delivery ratio, end-to-end delay, and parent change.

5.5 Proposed Objective Function for Energy Scavenging Nodes

The proposed optimized routing for energy scavenging nodes *rpl_QoSof* utilizes the combined design concepts of the available objective functions in COOJA simulator software of Contiki OS. The OF combines hop-count and ETX metric files which are already provided in the system for path and preferred parent selection within an RPL Instance. Similarly, the standardised objective functions use just one benchmark for metric consideration, because the default OFs use only one routing metric for path and preferred parent selection. However, this research will combine two metrics in this approach thereby increasing the benchmark to two instead of one. By doing this, this work simply reduces the frequency of handovers which causes unnecessary overhead and eventually improve network performance. The proposed routing metrics all work together within the objective function to enhance the path and preferred parent selection, making the objective function a requirement for energy scavenging in IoT applications. The integration of node and link metric creates the optimization needed for providing an adaptable solution for static energy scavenging routing protocol.

5.6 Link Quality Indicator (LQI)

LQI shows the quality of successfully received information, it gives the correlation value for every received packet using the length field of the PHY header together with PHY Service Data Unit (PSDU) (Bildea, 2013). The LQI range for CC2420 radios as discussed by (Lee & Chung, 2011) ranges from 0 to 255, but the reality is that there is no clear computation of LQI value as it remains hardware specific according to the hardware providers (de Waal & Gerharz, 2003).

RSSI and LQI are constantly computed for every received transmitted packet and their association for determining a good signal makes them highly important metrics in other to know good links in dynamic environments. An inherent characteristic of WSN devices is their susceptibility to fluctuations in link quality, its unreliable nature and at times frailness in connectivity. WSN use low power radios that are prone to interference, having multipath distortion and highly sensitive to noise, making the use of links with high reliability an important feature in the route selection process. Protocol reliability can be increased using an accurate link quality estimator (LQE) to determine efficient routes. The value of a link quality estimator indicates a precise point in time and does not provide any further information concerning the end-to-end link quality which is measured from the RSSI of LQI. Cross-layer information dissemination during route discovery process utilizes the LQE to determine the accuracy and reliability of links during this phase of route formation. Similarly, nodes in this scenario can continuously adapt to route selection process by estimating end-to-end link qualities, understanding network conditions, and are programmed to know the remaining energy of immediate neighbours.

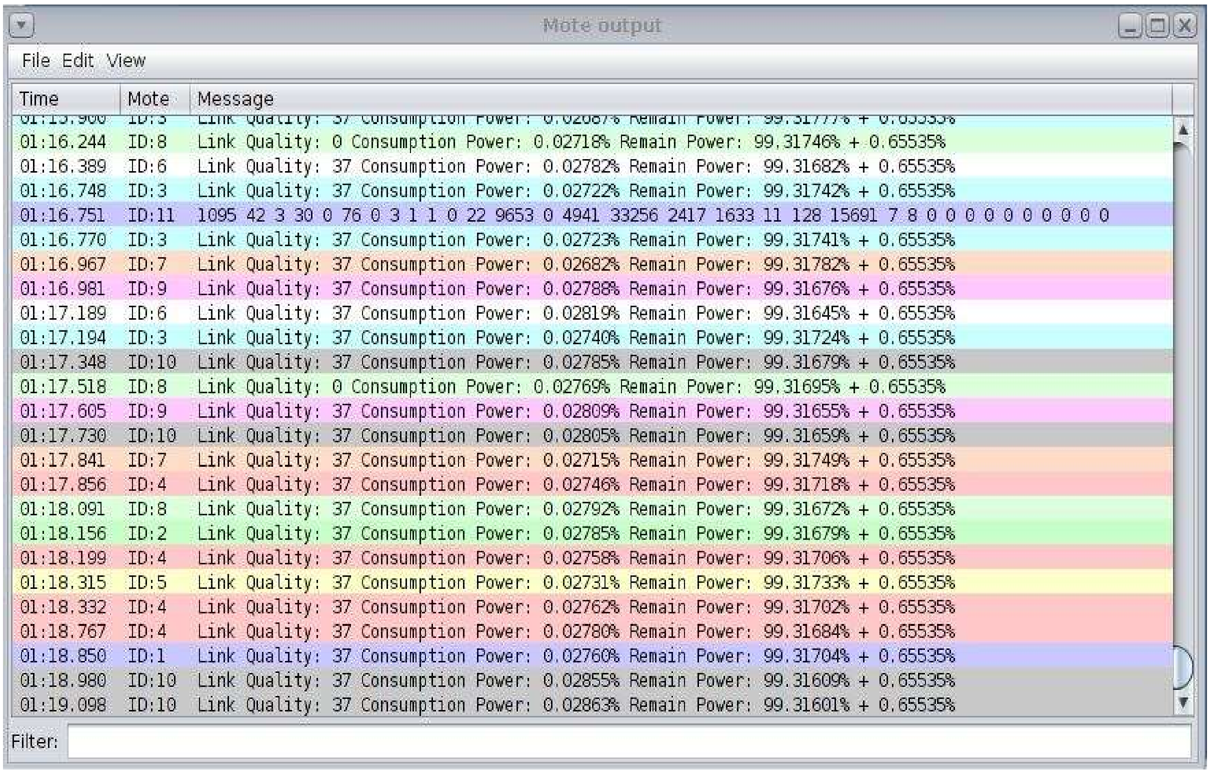
5.7 Simulation Setup, Results and Analysis

5.7.1 Simulation Setup

The enhanced energy efficiency routing protocol was implemented in COOJA simulator which is a wireless sensor network simulator embedded in Contiki Operating System 3.0. COOJA network simulator is an open-source software in Contiki OS that has all the detailed implementation files of RPL, with all source codes available making RPL modification possible and straightforward. The two objective functions available in COOJA simulator are OF0 and MRHOF which uses hop-count and ETX metrics respectively.

Two other routing metrics were introduced (remaining energy and link quality) which were combined with the standardised ETX in this chapter. The simulation was done using different scenarios with varying load densities which were used to test the performance of the new proposed objective function and compared with the standardised objective functions. Figure 5-3 shows the various performance of different nodes using the energy OF. The first parameter considered in this routing protocol is the energy consumption level of different nodes in the network as well as displaying the consumption power of the nodes. The second parameter considered is the link quality between each node to the sink node while the ETX value is only considered as a tie breaker. It will be noticed that this work created another 16-bit assigned integer, it is because the value 65535 is the maximum value and the research needed to reflect the exact energy value of nodes as the simulation progresses.

Similarly, this work tested using random positioning, although random distribution presented a greater challenge in uneven energy distribution among nodes due to unfairness in positioning of network nodes. Load balancing plays a big role in ensuring uniform energy distribution which can easily affect the network lifetime particularly with battery-operated nodes, hence some of the most important improvement consideration for this simulation will be the energy consumption level of nodes, packet delivery function and network lifetime.



```
01:16.244  ID:8   Link Quality: 0 Consumption Power: 0.02718% Remain Power: 99.31746% + 0.65535%
01:16.389  ID:6   Link Quality: 37 Consumption Power: 0.02782% Remain Power: 99.31682% + 0.65535%
01:16.748  ID:3   Link Quality: 37 Consumption Power: 0.02722% Remain Power: 99.31742% + 0.65535%
01:16.751  ID:11  1095 42 3 30 0 76 0 3 1 1 0 22 9653 0 4941 33256 2417 1633 11 128 15691 7 8 0 0 0 0 0 0 0 0 0 0
01:16.770  ID:3   Link Quality: 37 Consumption Power: 0.02723% Remain Power: 99.31741% + 0.65535%
01:16.967  ID:7   Link Quality: 37 Consumption Power: 0.02682% Remain Power: 99.31782% + 0.65535%
01:16.981  ID:9   Link Quality: 37 Consumption Power: 0.02788% Remain Power: 99.31676% + 0.65535%
01:17.189  ID:6   Link Quality: 37 Consumption Power: 0.02819% Remain Power: 99.31645% + 0.65535%
01:17.194  ID:3   Link Quality: 37 Consumption Power: 0.02740% Remain Power: 99.31724% + 0.65535%
01:17.348  ID:10  Link Quality: 37 Consumption Power: 0.02785% Remain Power: 99.31679% + 0.65535%
01:17.518  ID:8   Link Quality: 0 Consumption Power: 0.02769% Remain Power: 99.31695% + 0.65535%
01:17.605  ID:9   Link Quality: 37 Consumption Power: 0.02809% Remain Power: 99.31655% + 0.65535%
01:17.730  ID:10  Link Quality: 37 Consumption Power: 0.02805% Remain Power: 99.31659% + 0.65535%
01:17.841  ID:7   Link Quality: 37 Consumption Power: 0.02715% Remain Power: 99.31749% + 0.65535%
01:17.856  ID:4   Link Quality: 37 Consumption Power: 0.02746% Remain Power: 99.31718% + 0.65535%
01:18.091  ID:8   Link Quality: 37 Consumption Power: 0.02792% Remain Power: 99.31672% + 0.65535%
01:18.156  ID:2   Link Quality: 37 Consumption Power: 0.02785% Remain Power: 99.31679% + 0.65535%
01:18.199  ID:4   Link Quality: 37 Consumption Power: 0.02758% Remain Power: 99.31706% + 0.65535%
01:18.315  ID:5   Link Quality: 37 Consumption Power: 0.02731% Remain Power: 99.31733% + 0.65535%
01:18.332  ID:4   Link Quality: 37 Consumption Power: 0.02762% Remain Power: 99.31702% + 0.65535%
01:18.767  ID:4   Link Quality: 37 Consumption Power: 0.02780% Remain Power: 99.31684% + 0.65535%
01:18.850  ID:1   Link Quality: 37 Consumption Power: 0.02760% Remain Power: 99.31704% + 0.65535%
01:18.980  ID:10  Link Quality: 37 Consumption Power: 0.02855% Remain Power: 99.31609% + 0.65535%
01:19.098  ID:10  Link Quality: 37 Consumption Power: 0.02863% Remain Power: 99.31601% + 0.65535%
```

Figure 5- 2: Output Window for Energy OF

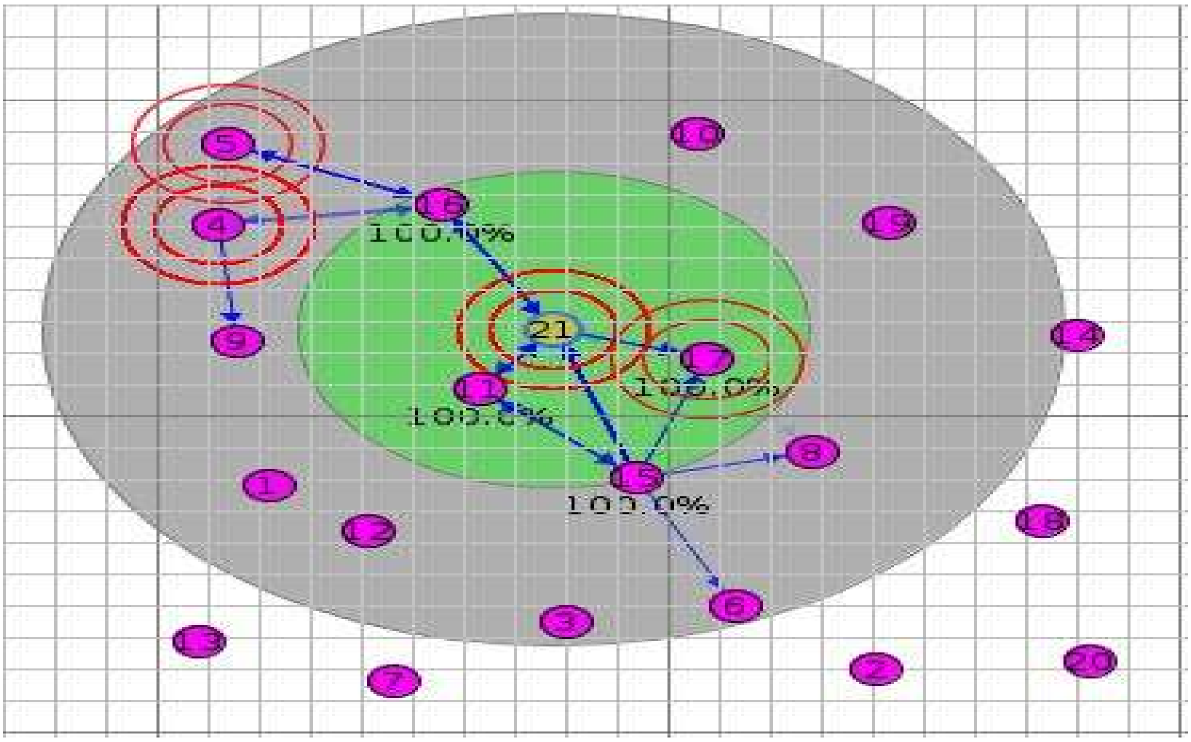


Figure 5- 3: Simulated Node arrangement for Energy Routing

This proposed enhanced energy efficiency objective function will only be simulated on battery operated devices not energy scavenging nodes (i.e. nodes using irregular means of energy sustenance like solar or wind turbines or mains powered).

Table 5-1 gives the various specifications utilized for running the *rpl_energyof* in COOJA simulator, this work used different node densities from 10, 20 and 30 nodes with 1 sink node in a 200m × 200m simulation area as displayed vividly in Figure 5-4. The yellow node with number 21 is the sink node where all traffics are directed toward, while the scattered red nodes numbered 1-20 are the randomly positioned sender nodes.

Table 5- 1: Simulation Parameters

Parameters Used	***Value Specification***
Network Simulator	*COOJA in Contiki OS 3.0*
Simulation Area	*200m × 200m*
Number of Nodes	*10,20,30 mobile nodes + 1 sink node*
Routing Protocol	***rpl_energyof***
Emulated Nodes	*Z1 Mote*
Transmission Range	*50m*
Simulation Time	*1800 seconds*
Node Deployment	*Random*
MAC layer protocol	*ContikiMAC/6LowPAN*
Radio Environment	*Unit Disk Graph Medium (UDGM)*
Radio	*CC2420*

In this scenario of enhanced energy efficient routing in IoT, nodes are not mobile but this work still implemented the OF using mobile nodes with the same mobility models in chapter 4 to test its performance in the event of energy saving routing with mobile nodes on *rpl_adaptiveof* in section 7.4 which will be discussed in much details later. This research utilized the UDGM radio environment with CC2420 radio for IEEE 802.15.4 on Contiki MAC protocol, where simulations were run for 1800 seconds and energy measurement for different nodes were recorded over time particularly those nodes in one hop neighbour of the sink node.

Similarly, this work tested using random positioning of nodes, although it did not pay attention to the energy distribution among nodes where load balancing is not a criterion in this type of routing as nodes are believed to be energy scavenging i.e. having a means of energy sustenance like solar or wind or mains-powered. This proposed optimized routing protocol for energy scavenging devices will be simulated with the assumption that devices always have 100% energy, thereby ignoring the aspect of energy consumption. Also, it is assumed that devices are well placed within radio ranges of one another with an adequate link quality between them.

The various specifications utilized by *rpl_QoSof* in COOJA simulator is indicated at Table 6-1, this work also utilized different node densities from 10,20 and 30 nodes with 1 sink node in a 200m × 200m simulation area. In this scenario, this work has not implemented mobility hence no mobility model was utilized, this work utilized UDGM radio environment with CC2420 radio for IEEE 802.15.4 on Contiki MAC protocol with simulations run for an hour.

5.7.2 Simulation Results

This work tested the performance of *rpl_energyof* with certain metrics which allows us to understand the efficiency and effectiveness of the modified OF towards the performance of the network. These metrics are energy consumption, packet delivery function, parent change and end-to-end delay, where each of these metrics have been clearly defined and explained in the previous chapters. Figure 5-4 indicates the packet delivery ratio which is popularly referred to as the PDR measured in percentage, *rpl_energyof* shows a very high packet delivery rate of 92.80%, 89% and 86% for 10, 20 and 30 node distribution respectively which is almost 15% higher than the standardised versions of RPL OF when deployed on static nodes.

The design and implementation of *rpl_energyof* is very optimistic in the sense that it uses three routing metrics for decision making, these metrics are link quality, remaining node energy and ETX, and they all play a vital role in making suitable routing decisions. Standardised version of RPL OF shows a much lower packet delivery rate of almost 35% lower than the OF, and *rpl_mrhof* shows better performance in terms of PDR than *rpl_OF0* for static node settings. Additionally, Figure 5-5 indicates the average energy consumption of nodes after the simulation was run for an hour, also *rpl_energyof* showed far better performance than the standardised versions. Mainly because the approach tends to utilize the remaining energy of a node as the main metric for making routing decisions while also considering the link quality of nodes, the optimization of these metrics ensures best routes are always selected.

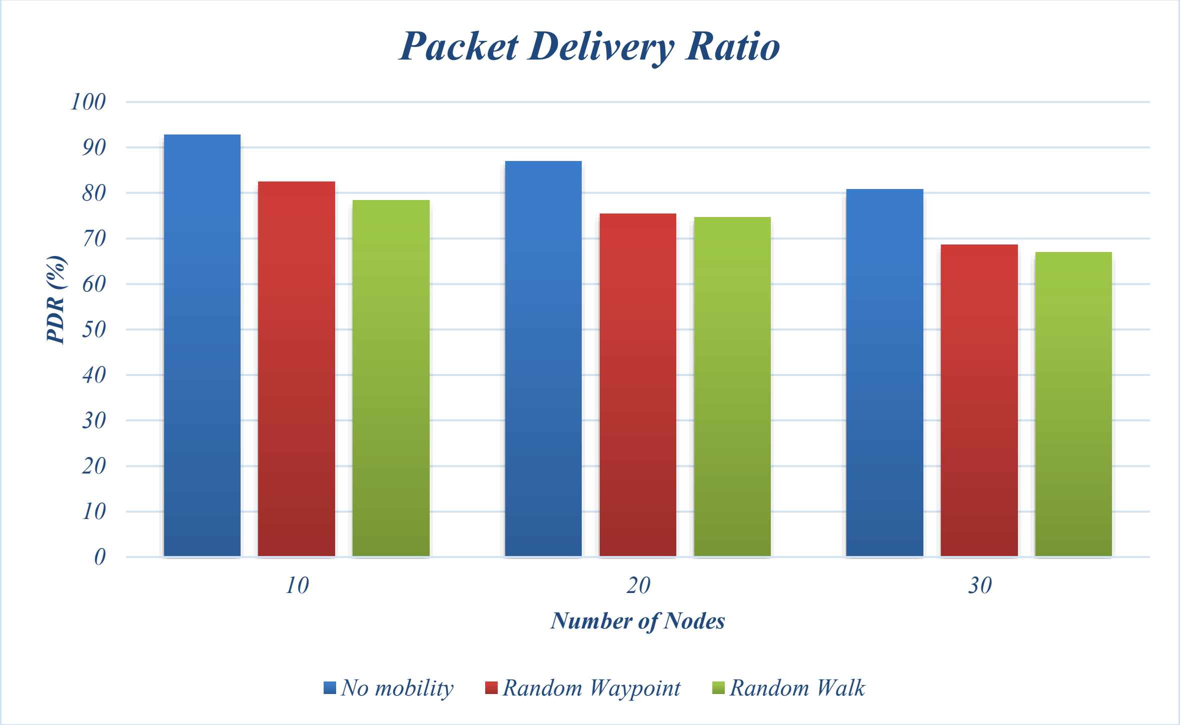


Figure 5- 4: PDR for *rpl_energyof*

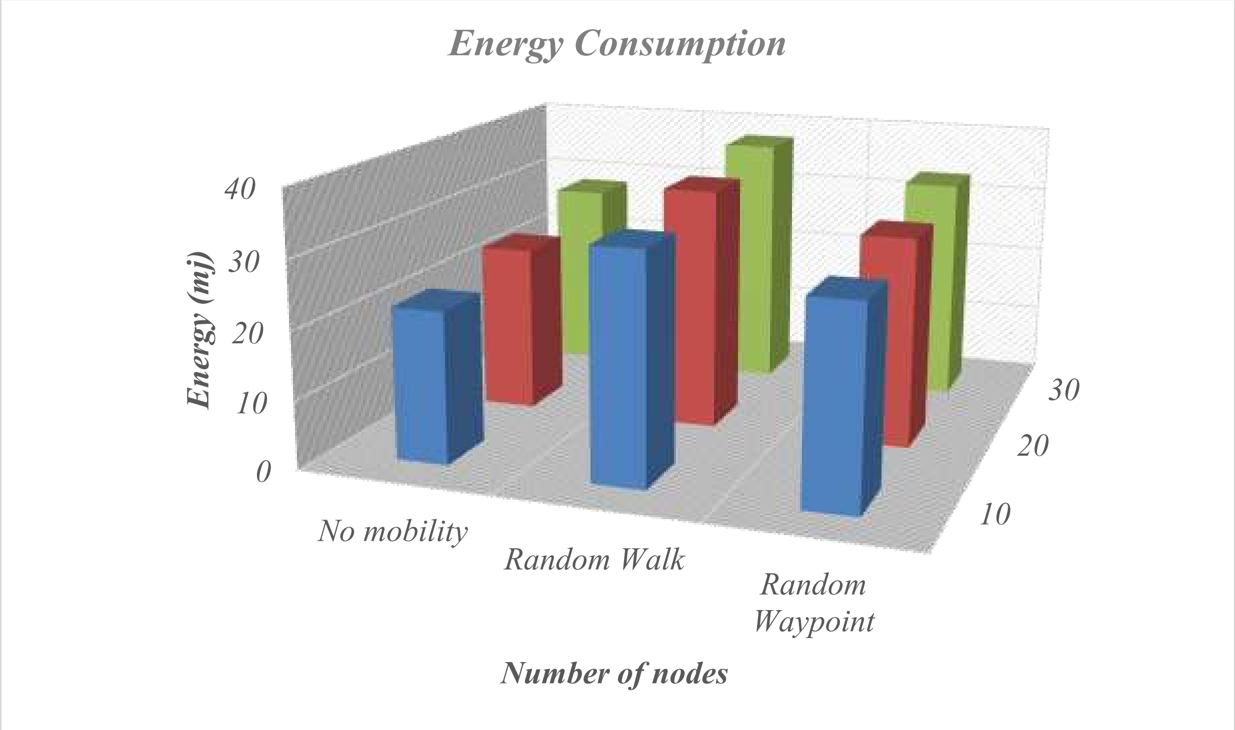


Figure 5- 5: Energy consumption for *rpl_energyof*

Similarly, optimized selection of routes leads to low packet loss as the rate of retransmission, broken and failed links are greatly reduced. Similarly, this work evaluated the path and preferred parent change for the energy OF, Figure 5-6 indicates the number of times a node switches parent while transmission is on-going.

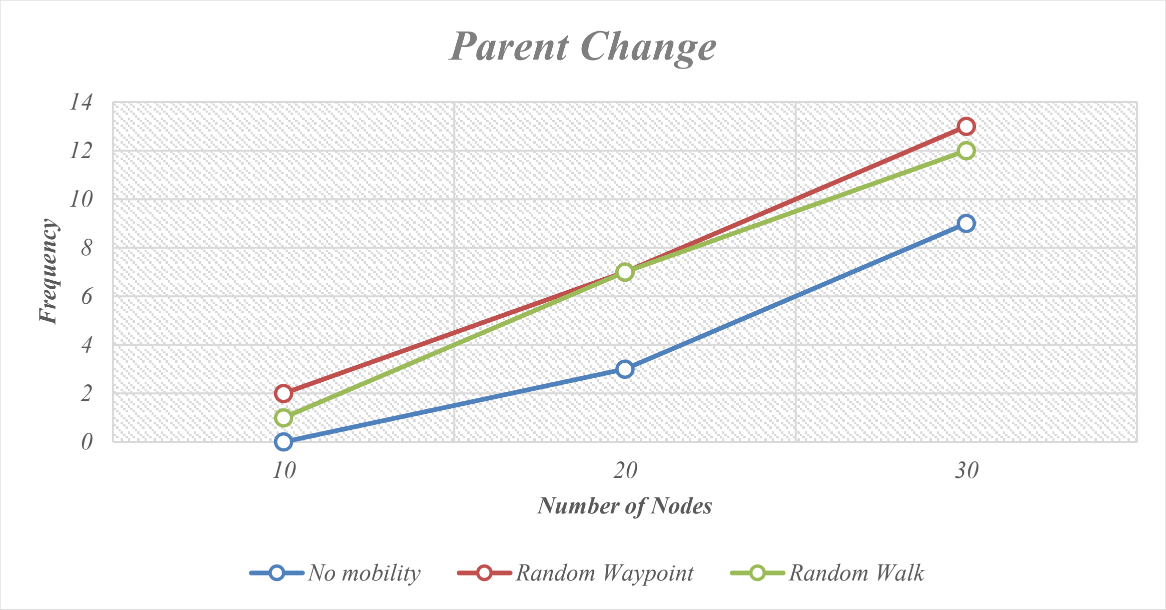


Figure 5- 6: Parent Change for *rpl_energyof*

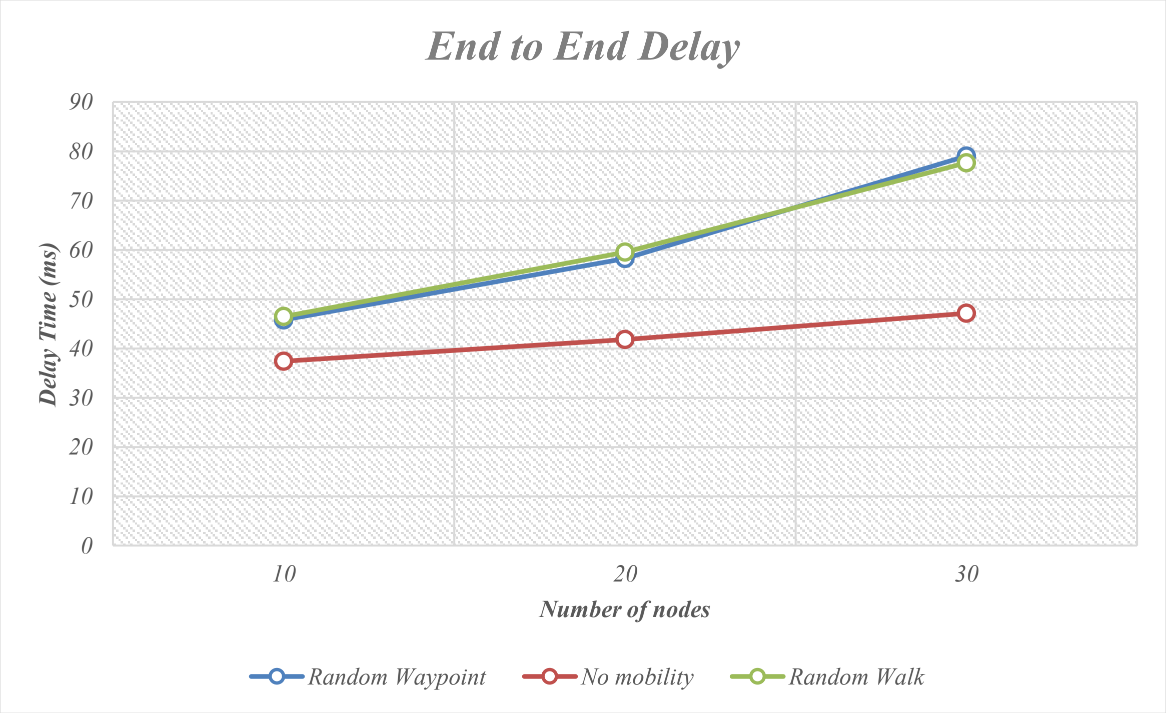


Figure 5- 7: End-to-end Delay for *rpl_energyof*

The approach incurs the highest number of parental changes as compared to the standardised versions in static and low node densities, because the OF continuously determines the most optimal path and preferred parent at every iteration. Also, this research changed the *PARENT_SWITCH_THRESHOLD* value from one to two meaning that two or more metrics will be used for path and preferred parent switch as opposed to one. Again, with the energy metric, a nodes energy consumption rate will be measured before been chosen as a potential path or preferred parent selection. Figure 5-7 shows the latency (end-to-end delay) as it increases with increasing number of nodes, where *rpl_energyof* still showed a superior performance to the standardised version of the protocol.

The OF also plays a vital role in path and preferred parent selection since they all use different metric for route and preferred parent selection, the approach proposes a more robust and scalable method in this regard. End-to-end delay tends to be slightly higher with increasing number of nodes which often leads to increase in number of hops but having no impact on throughput and packet delivery. The end-to-end delay is measured as an average for all the different node densities and OFs.

This work simulated the performance of rpl_staticof and measured the performance with the standardised versions of the RPL OF using certain metrics so that it can ascertain areas of improvement. These metrics are packet delivery function, overhead and end-to-end delay, each of these metrics have been clearly defined and explained in the previous chapters.

Figure 5-8 shows the PDR value in percentage, which signifies the number of successfully transmitted versus the total number of packets sent in the network. *rpl_QoSof* shows better performance when implemented in a static environment as compared to the standardised versions of the RPL OF. *rpl_QoSof* showed an average PDR of 94.06%, 89.90% and 84.76% for 10, 20 and 30 nodes respectively. The static OF shows high level performance when implemented on a static environment and there is no consideration for mobility, energy consumption or parent change, as nodes are carefully placed within ranges of one another with no extra layer of processing required. The approach shows better performance in PDR as compared to the standardised versions of the RPL protocols under mobility at the same transmission rate with very low number of dropped packets.

Similarly, Figure 5-9 displays the average end-to-end delay of the of static OF as compared to standardised versions, the OF shows better performance due to its optimal path and preferred parent selection.

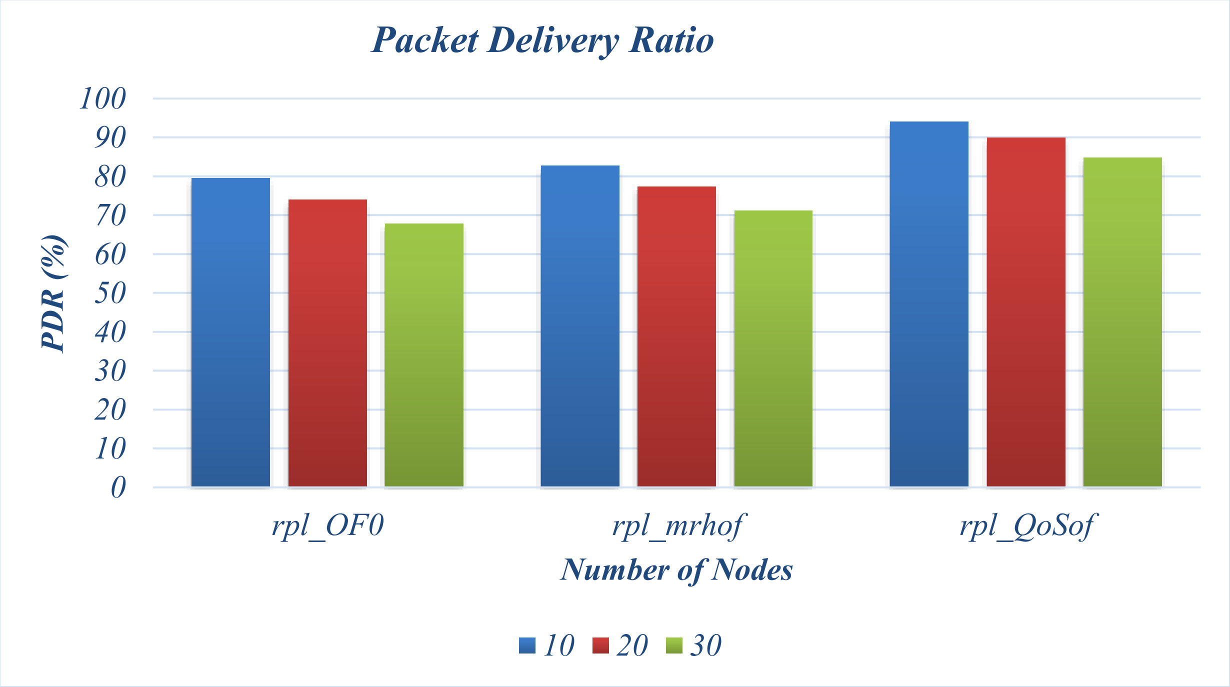


Figure 5- 8: PDR for *rpl_QoSof*

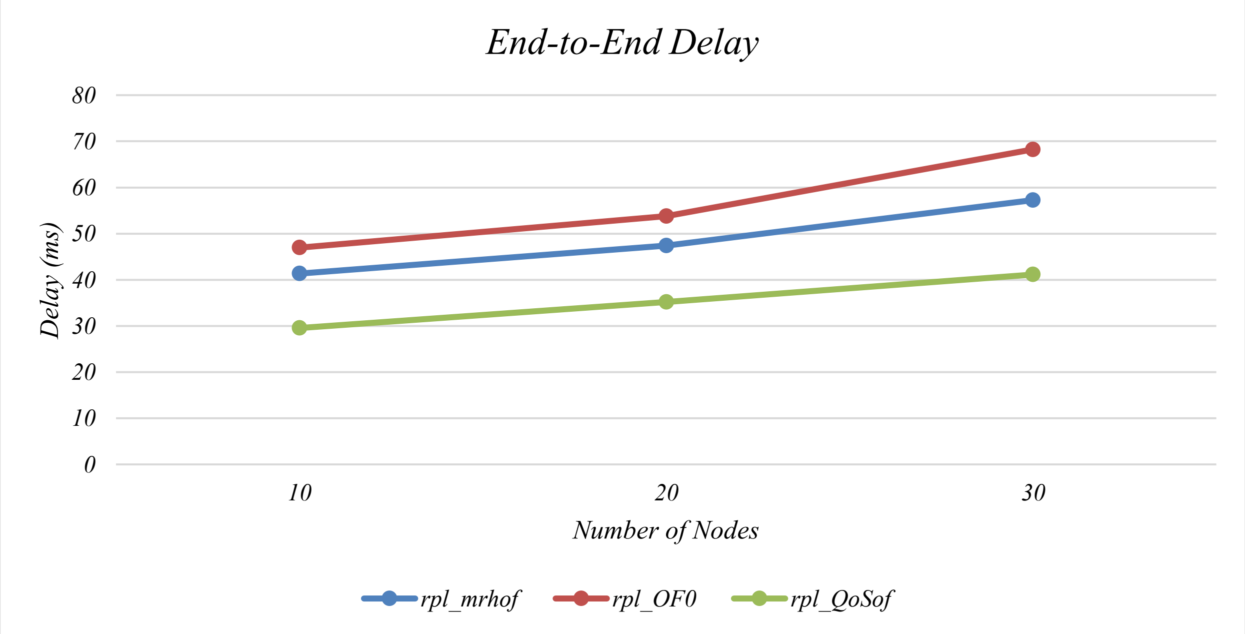


Figure 5-9: End-to-End Delay for *rpl_QoSof* and Standardised Versions

This is because in static approach, nodes are carefully arranged within radio ranges and having very strong link quality to reduce delay especially the queue and processing delays, thereby freeing the buffer occupancy from storing too many packets.

5.8 Summary

This chapter discussed the performance evaluation of *rpl_energyof* as compared with standardised versions of RPL OF for enhanced energy saving routing in both static and dynamic IoT applications by providing reliable and efficient algorithms. Generally, *rpl_energyof* showed improved performance overall when compared with standardised versions of the RPL OF in terms of end-to-end delay, packet delivery and energy consumption, except for parent change. The approach achieves an average of 92% PDR, 37.39ms latency and 22.30mj energy consumption as compared to the standardised version of RPL. Simulation results indicates how the energy OF impacts the entire performance of RPL protocol in enhancing energy saving. The approach relies heavily on the remaining energy of nodes which is a major metric consideration in enhanced energy saving routing, it is done in conjunction with other routing metrics link quality and ETX for path and preferred parent selection. The optimization of the three combined metrics in the objective function was crucial for improved routing performance to achieve energy efficiency.

Chapter Six

Adaptive Objective Function for RPL Protocol in IoT (Optimum Reliable Objective Function OR-OF)

6.1 Introduction

There has never been a single routing protocol in WSN that can suit different application needs because every application has a particular routing protocol that can satisfy its needs. Some applications are highly mobile and require a routing protocol that supports mobility, others require QoS making the need for high packet delivery of paramount importance, and other applications require prolonged network lifetime. Additionally, it is very difficult if not impossible to have a single routing protocol that will guarantee robustness, scalability, context awareness, energy efficiency, proactive route maintenance, reduced end-to-end delay, mobility support, irregular node deployment or topology changes, congestion control, QoS and high throughput. To this regard, this work proposes an adaptive routing protocol that supports mobility, energy efficiency and energy scavenging nodes, and its performance is evaluated against some popular routing metrics which include PDR, energy consumption, end-to-end delay, throughput, and parent change.

In this chapter, this work proposed an adaptive objective function in RPL protocol (Optimum Reliable Objective Function) which combines four implementations of OFs behaving as a single OF that suits different routing requirements for different applications. The novelties of this chapter are highlighted as thus: (i) The creation of an adaptive routing protocol for RPL using fuzzy logic design. (ii) The combination of six different routing metrics (both node and link) in the same OF used for different purposes. (iii) Using Received signal strength indicator to determine the mobility conditions or otherwise of different nodes. (iv) Creates an optimum reliable transmission by switching to different objective functions with negligible delay. (v) Ability to adapt to different routing situations and varying node densities. (vi) Implementation of more than one routing protocol participating independently of one another in the same DODAG of an RPL Instance.

The remaining sections are organised as thus: Section 6.2 and 6.3 explains fuzzy logic and the proposed fuzzy logic based objective function, section 6.4 describes the proposed protocol design. Section 6.5 explains the protocol implementation including the adaptive algorithm, Section 6.6 describes the simulation setup, results, and analysis as it is compared with the standardised RPL objective functions and section 6.7 gives the proposed mathematical model to be utilized. Lastly, section 6.8 summarizes the chapter.

6.2 Fuzzy Logic (FL)

Using fuzzy logic in routing algorithms have proven to be an effective approach in wireless sensor networking as well as other related disciplines such as robotics, speech recognition, signal processing, distributed computing and many other business and marketing related fields. Fuzzy logic has shown great potentials in WSN particularly on the need for combining different parameters that allows for effective and efficient evaluation of various application needs with emphasis on the different layers of the OSI reference model. The conceptual logic of fuzzy reasoning as originally proposed by a scientist known as Lofti-Zadeh cited in (G. Li, Ma, Liu, & Shu, 2017) to mimic human thought and decision-making capabilities offering unique solutions to several control problems.

The inherent nature of FLs uncertainty and ambiguity gives it the capacity to handle conflicting, imprecise information and multiple combination of input parameters passing through the rule-based and inference engine to produce a single output metric. It can also process non-linear input variables with merits of robustness and ease of implementation where complex problems are analysed using simple linguistic variables. In WSN, it shows more potential of improving routing performance schemes, clustering and cluster head selection, data aggregation and others. It is a fact that computers require a lot of details in other to reason like a human being, hence, adopting human reasoning will entail dealing with the uncertainties and ambiguities existing in the real world such as natural language processing (Zuo, 2013).

6.3 Proposed Fuzzy Logic Based RPL (*rpl_adaptiveof*) Objective Function

RPL is a proactive routing protocol which means that it discovers and maintains routes immediately the routing protocol is initiated, path and preferred parents that satisfy the condition imposed by the OF are collected and stored. This approach utilizes a mechanism for identifying a threshold which is used to determine the nature of routing to be utilized at every point in time. The metric combination to be used is dependent on the type of routing protocol chosen and it is triggered only by changes in routing priority. The standardised RPL has two

OFs *rpl_OF0* and *rpl_mrhof* which uses hop-count and ETX metrics as extensively discussed in the previous chapters, this work then proposed the three OFs *rpl_energyof*, *rpl_mobilityof* and *rpl_QoSof*, using several combinations of metrics.

However, in this design, this work tries to exploit the concept of fuzzy logic to combine all the three proposed OFs into a single adaptive OF that can suit different application needs. The combination of different metrics in making routing decisions reduces the frequency of unwanted handovers and the switch between different preferred parents which overall improves the efficiency of the routing protocol and network performance. Within each proposed OF, the path and preferred parent switch threshold has been changed to two, this means that within an OF at least two metrics must be utilized to influence any switch.

This proposed *rpl_adaptiveof* tends to determine and switch between OFs in real time depending on different situations, for instance, when the adaptive OF is initiated, the algorithm transmits every node's remaining energy, RSSI reading and its connectivity status (i.e. for energy scavenging nodes), when a node in a network suddenly starts moving, the RSSI readings changes from the previously recorded reading and it automatically switches the routing protocol of that node to *rpl_mobilityof* which best suits that particular situation in time. The integration of different OFs in RPL into a single OF creates a certain level of optimization making this approach an adaptable and better solution for different IoT application requirements.

6.4 Proposed Protocol Design

The Fuzzy Logic Adaptive OF proposed in this research jointly considers remaining energy, latency, link quality, ETX, RSSI readings and hop-count as inputs while categorizing them into their different categories and finding the appropriate balance between them. Most of the metrics utilized were not used as constraints and minimum value is selected for most of the metrics except for remaining energy and link quality. Using the triangular membership function which maps various input variables into the fuzzy sets within the FLS, and values used within the thesis are Low, Medium, and High for all the fuzzy inputs described. To obtain a suitable outcome for the combination of different levels of input, this work defined a more normalized generic triangular membership function for all the input metrics mostly having the range of [0, 1].

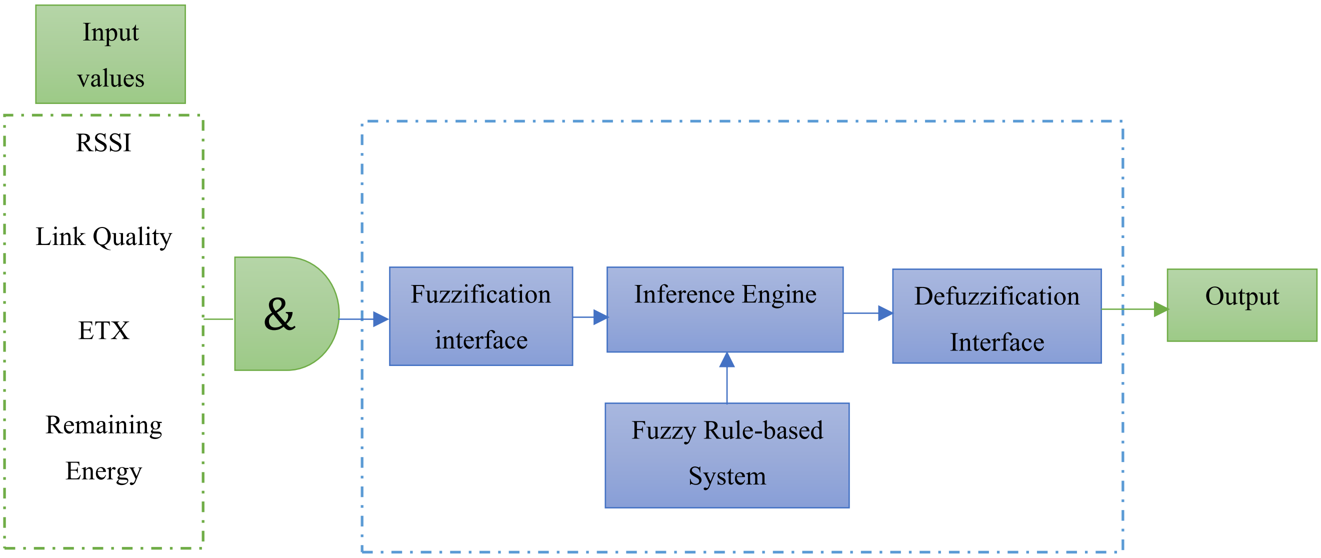


Figure 6- 1: Fuzzy Logic System

The membership function can be expressed as follows, $F_{Low}\ (x), F_{Medium}\ (x), and\ F_{High}\ (x)$, defined by the normalized fuzzy set and it can be represented graphically and mathematically as follows (Zuo, 2013);

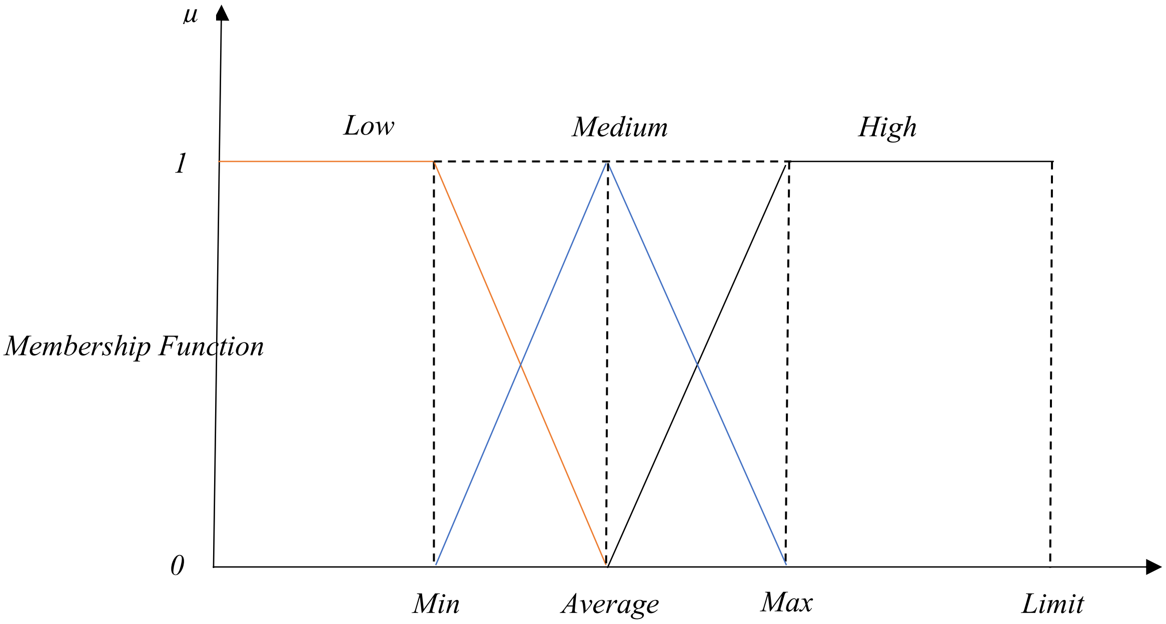


Figure 6- 2: The Generic Triangular Membership Function Graph

$$F_{Low}\,(x) \;=\; \begin{cases} 1, & 0 \leq\, x \,\leq\, Min \\ \dfrac{Average \,-\, x}{Min} & Min \,<\, x \,\leq\, Average \\ 0, & Average \,<\, x \,\leq\, Limit \end{cases} \tag{6.1}$$

$$F_{Medium}(x) = \begin{cases} 0, & 0 \leq\, x \,\leq\, Max \\ \dfrac{x \,-\, Min}{Min}, & Min \,<\, x \,\leq\, Average \\ \dfrac{Max \,- x}{Average}\,, & Average \,<\, x \,\leq\, Max \\ 0, & Max \,<\, x \,\leq\, Limit \end{cases} \tag{6.2}$$

$$F_{Max}\,(x) \;=\; \begin{cases} \dfrac{x \,-\, Average}{Average}\,, & Average \,<\, x \,\leq\, Max \\ 1, & Max \,<\, x \,\leq\, Limit \end{cases} \tag{6.3}$$

The membership function defined in this thesis are low, medium, and high for all the input values in the fuzzification aspect. The inference rule utilizes the IF-THEN rule to map out various fuzzy inputs with the likely fuzzy outputs with all the levels clearly defined as explained in Table 6-1. The FLS within the inference engine shows all the likely combination of possible outcomes from each level of the metric arrangements. An IF-THEN example utilized in this work is given in Table 6-1. The FLS inference rule utilized in this thesis for enhanced energy and mobility routing is specified in the table above with a minimum of 18 rules. The minimum and maximum values are defined on the triangular membership function which is used to map a certain input to a predefined output variable.

6.5 Protocol Implementation

Like the previous implementations of the proposed OFs, the adaptive OF was also implemented in COOJA with a mobility plugin and incorporated with Bonnmotion tool for mobility scenario implementation. Although this work only utilized the RWP mobility model for simulating mobile nodes in this scenario, while all other specifications remain the same as that of *rpl_mobilityof.*

While implementing the adaptive approach two methods were realized: the first method requires user intervention, meaning the user will be the one to determine when to switch between the proposed objective functions based on some predefined thresholds, these thresholds can be summarized below;

Table 6- 1: Inference rule in the Fuzzy Logic System

No.	**ETX**	**Energy**	**Link Quality**	**Route Suitability**
1.	Low	Low	Low	**Average**
2.	Low	Low	High	**Very Good**
3.	Low	Medium	Low	**Average**
4.	Low	Medium	High	**Very Good**
5.	Low	High	Low	**Bad**
6.	Low	High	High	**Bad**
7.	Medium	Low	Low	**Average**
8.	Medium	Low	High	**Very Good**
9.	Medium	Medium	Low	**Average**
10.	Medium	Medium	High	**Average**
11.	Medium	High	Low	**Bad**
12.	Medium	High	High	**Bad**
13.	High	Low	Low	**Average**
14.	High	Low	High	**Average**
15.	High	Medium	Low	**Bad**
16.	High	Medium	High	**Bad**
17.	High	High	Low	**Bad**
18.	High	High	High	**Bad**

a. For Enhanced Energy Saving Routing (*rpl_energyof*)
    i. If the average remaining energy of the entire network is less than 50%.
    ii. If a packet has no other route but the route with the least energy (less than 50%) or packet TX rate is high.
    iii. If the average energy consumption of the network is higher than a certain benchmark.

iv. When the intention of routing is based on energy saving as devices are expected to run for a long time or devices will transmit high number of packets or otherwise.

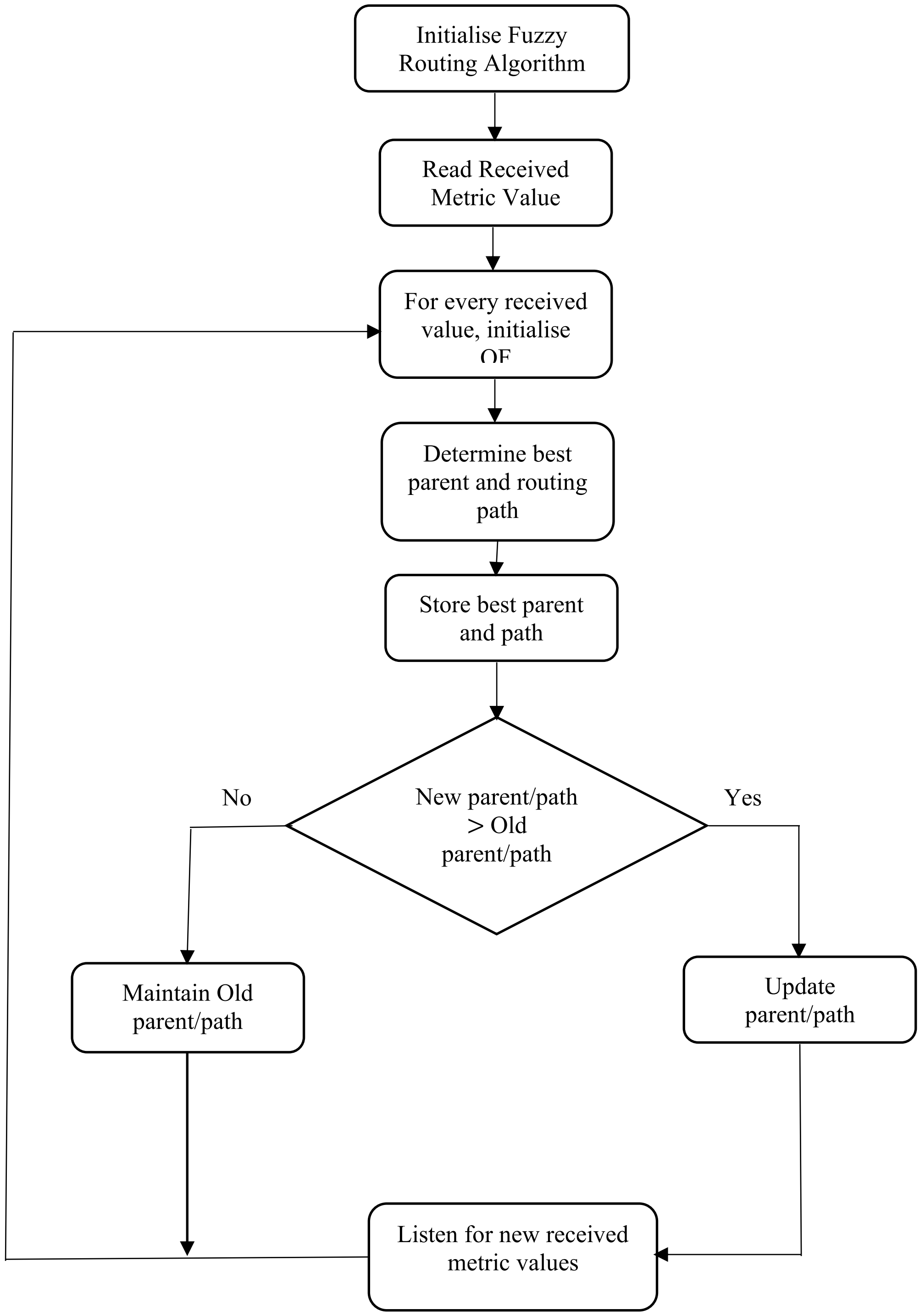


Figure 6- 3: Process Flow for Fuzzy Logic OF

Algorithm 6- 1: Adaptive OF for IoT

```
Algorithm
Begin: Initialise Algorithm
Initialise Fuzzy Logic
mf_type* fuzzification()
rule_evaluation()
Compute_degree_of_membership (mf_type *mf, int input)
defuzzification()
Int Compute_routing_priority;
        {
        Int m_RemainingEnergy = mf → value;
        Int m_rssi = mf → point1;
        Int m_ConnectedPowerSource = mf → point2;
                If (m_RemainingEnergy > 0 && m_RemainingEnergy < 50);
                        {
                        return ENERGY_ROUTING;
                        }
                If ((m_rssi  != m_PrevRssi) && (m_PrevRssi < 0 ));
                        {
                        m_PrevRssi = m_rssi;
                        return MOBILITY_ROUTING;
                        }
                        m_PrevRssi = m_rssi;
                If (m_ConnectedPowerSource == 1)
                        {
                        return STATIC_ROUTING;
                        }
                return NORMAL_ROUTING;
                }
m_Priority = Compute_routing_priority (mf);
setPriority (m_Priority);
        If (m_Priority == ENERGY_ROUTING);
                {
                        new_metric = combination of (New_etx, Energy_metric, Lq_metric);
                }
        Else If (m_Priority == MOBILITY_ROUTING);
                {
                        new_metric = combination of (rssi, latency, hop_metric);
                }
        Else If (m_Priority == STATIC_ROUTING);
                {
                        new_metric = combination of (New_etx, hop_metric);
                }
        Else If (m_Priority == NORMAL_ROUTING);
                {
                        new_metric = New_etx;
                } /* update the link metric for this nbr*/
        nbr → link_metric = new_metric;
End
```

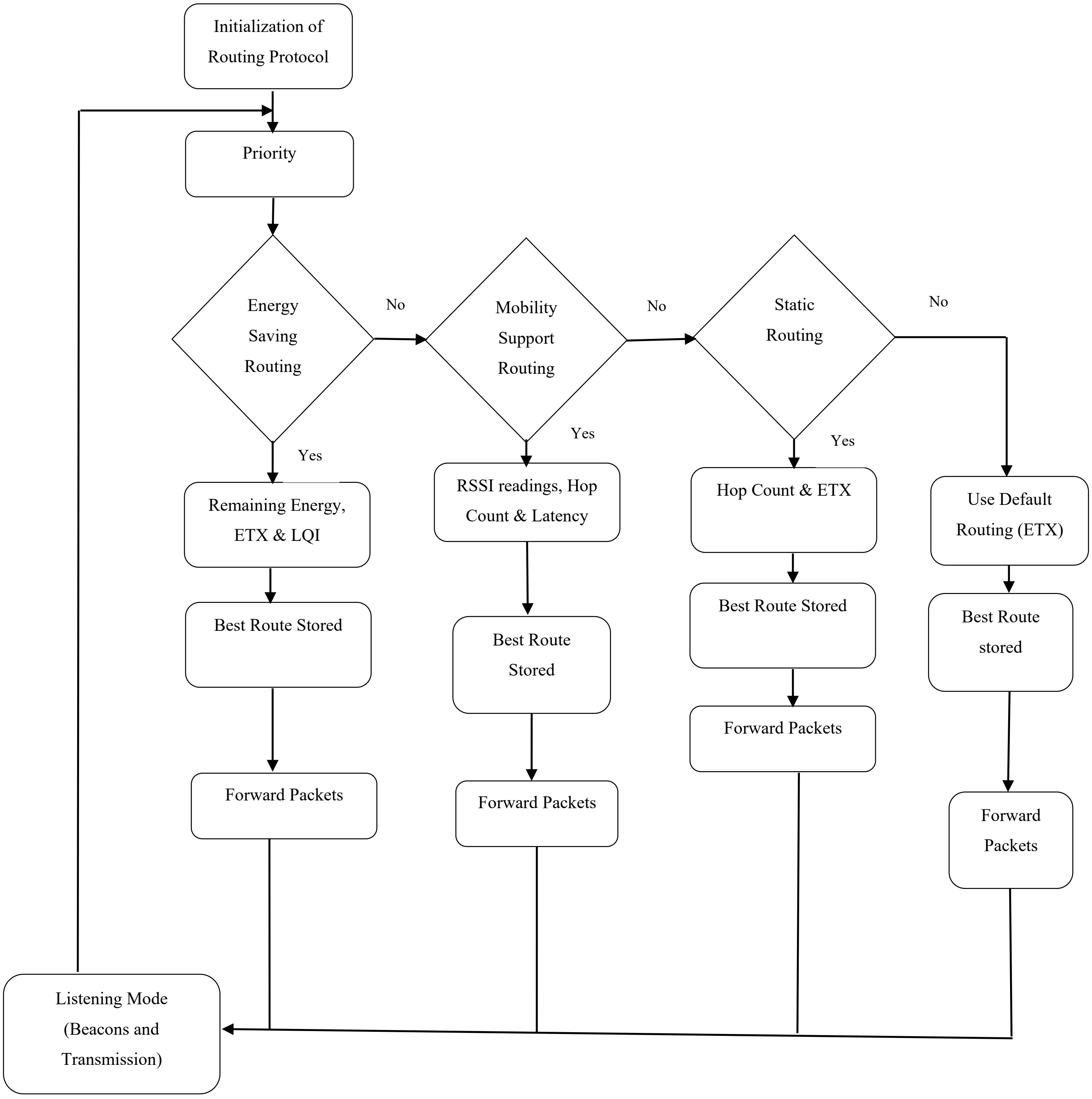


Figure 6- 4: Process Flow for Adaptive Routing Protocol

a. For Enhanced Mobility Support Routing (*rpl_mobilityof*)
    v. If a nodes RSSI value changes from last received (meaning a node is moving)
    vi. If a node changes path or parent more than 3 times with both node and link disjoint.
    vii. When the intention of routing is based on mobility models as defined by the user.

b. For Optimized Protocol for Energy Scavenging Nodes (*rpl_staticof*)
    viii. If nodes are always connected to a power source (either mains or energy scavengers e.g. solar) making energy level 100%.
    ix. When the intention of routing is based on static objective function which is the combination of both OF0 and MRHOF (hop-count and ETX).

c. For Normal Routing (*rpl_mrhof*)
    x. Used as the default routing protocol in this design (ETX).

The above threshold specification was designed to be user enabled, which means that the user or the network administrator will be the only one to determine when to switch between the OF based on the information provided above. The second method is the adaptive approach which is completely determined by the system based on the specified thresholds. The first of each threshold categories above are the ones utilized by *rpl_adaptiveof* in this work, for instance, for enhanced energy saving routing the threshold considered is the energy level of nodes (below 50%), for enhanced mobility support routing the threshold considered is changes in RSSI readings, for energy scavenging routing the threshold considered is the energy level (usually full or 100%) and for any other routing aside from the ones specified is the default routing protocol.

All these OFs were implemented using Fuzzy Logic which allows us to combine both the different OFs and metrics from various inputs into a single usable output at every time. However, after several tries on implementing all these thresholds in the adaptive OF, this work was not able to achieve all, only the first threshold in each category was implemented in this thesis, the rest were left for future research. The algorithm for the implementation of the adaptive OF is given in Algorithm 6-1.

6.6 Simulation Setup, Results and Analysis

6.6.1 Simulation Setup

The Adaptive FL routing protocol implemented in this research (OR-OF) was implemented in COOJA simulator in Contiki Operating System 3.0. COOJA network simulator is an open-source software in Contiki OS that has all the detailed implementation files of RPL with all source codes available making RPL modification easy and straightforward. The two objective functions available in COOJA simulator are OF0 and MRHOF which uses hop-count and ETX metrics respectively. This section presents an adaptive approach which combines four different objective functions behaving as a single OF while utilizing the concept of FL, the OFs are *rpl_energyof* for energy efficiency routing (with combined metrics of remaining energy, link quality and ETX), *rpl_mobilityof* for mobile nodes (with combined metrics of RSSI readings, hop-count, and latency), *rpl_QoSof* for energy scavenging nodes (with combined metrics of ETX and hop-count) and *rpl_mrhof* as default OF. The simulation was done using different scenarios with varying load densities which were used to test the performance of the new proposed objective function and compared with both the standardised objective functions and other proposed OF within this research work.

This research ran the simulation using random positioning of nodes with varying node densities and one sink node. All the OF were implemented in the adaptive approach with different node densities having few numbers of nodes implemented with mobility in RWP model while the other nodes are static nodes, with a static sink node. The modified RPL OFs were tested and presented negligible delay in switching between the different OF specified within the routing protocol, although some of the nodes presented irregular RSSI readings mostly caused by the COOJA module and were assumed to be mobile where the mobility OF was applied, this was referred to as the false positive factor which was outlined as one of the drawbacks of this method at Chapter 7.

But in general, the performance of the adaptive OF has been very satisfactory as all the scenarios involving the different proposed OF were utilized. The UDGM radio environment shows both the interference and transmission ranges of a simulated node and it also helps to know the exact position of nodes in the simulation environment. Table 6-1 gives the various specifications utilized for running the *rpl_adaptiveof* in COOJA simulator, this work also used different node densities from 10, 20 and 30 nodes with 1 sink node in a 200m × 200m simulation area as displayed vividly in Figure 6-4, while some other nodes were made to be mobile using the RWP mobility model.

The yellow node with number 6 is the sink node where all traffics are directed toward, while the scattered red nodes numbered 6-26 are the randomly positioned sender nodes and the purple nodes labelled 1-5 are mobile nodes. In this scenario of adaptive OF (OR-OF) routing, some nodes are mobile hence there is an implementation of RWP mobility model to those specific nodes, while the remaining nodes use the other proposed OF as the situation requires. This work utilized the UDGM radio environment with CC2420 radio for IEEE 802.15.4 on Contiki MAC protocol with 6LowPAN adaptation layer, where simulations were run for 1 hour and several performance metrics were recorded over time particularly those nodes in one hop neighbour of the sink node.

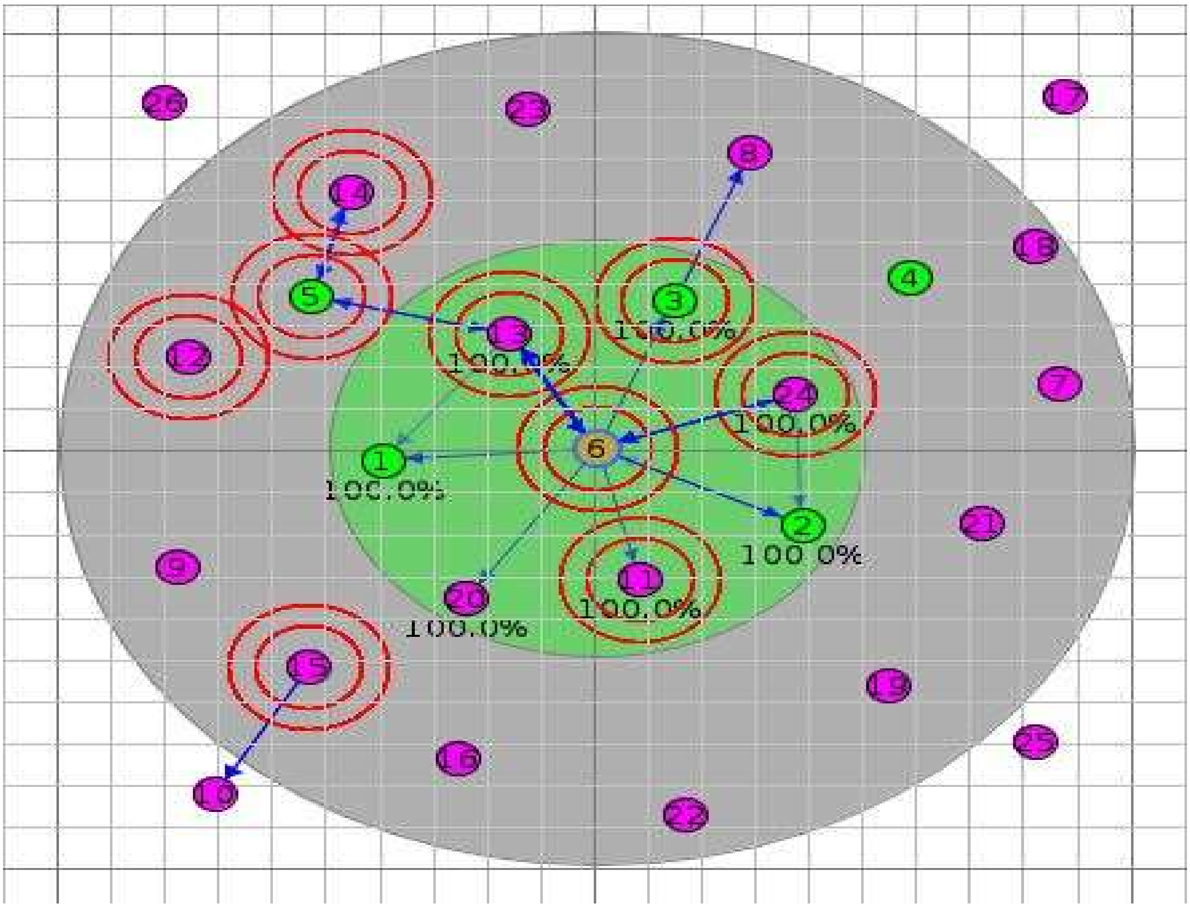


Figure 6- 5: Simulated Node Deployment

From the results of the simulation displayed in Figure 6-5, node 2 (ID:2) started moving immediately the *rpl_adaptiveof* is initiated thereby displaying its status with respect to its remaining energy, RSSI readings and power connectivity status which are displayed as 99, -82dBm and 0 respectively. The simulation mote output window has three columns representing the time, mote ID and message, the time indicates the exact moment a node transmits its message with the node ID number and the message which it displays.

Table 6- 2: Simulation Parameters

Parameters Used	***Value Specification***
Network Simulator	*COOJA in Contiki OS 3.0*
Simulation Area	*200m × 200m*
Number of Nodes	*10, 20 and 30 (with each having random number of mobile nodes) + 1 sink node*
Routing Protocol	***rpl_adaptiveof***
Emulated Nodes	*Z1 Mote*
Packet Size	*64 Bytes*
Number of Packets Transmitted	*3 Packets/Second*
Transmission Range	*50m*
OF utilized	*rpl_energyof, rpl_mobilityof, rpl_QoSof and rpl_mrhof*
Start-up Delay	*2000ms*
I_{min} / $I_{doubling}$	*13 / 6*
Simulation Time	*1,800 seconds*
Node Deployment	*Random Positioning*
Radio Environment	*UDGM (Unit Disk Graph Medium)*
MAC/ Adaptation Layer	*Contiki MAC/6LowPAN*
Radio	*CC2420*
Bonnmotion Tool Parameters for Mobility OF	
Minimum speed	*1m/s*
Maximum speed	*4m/s*
Minimum pause time	*2s*
Maximum pause time	*10s*
Mobility model	*Random Waypoint*
Interference range	*50m*

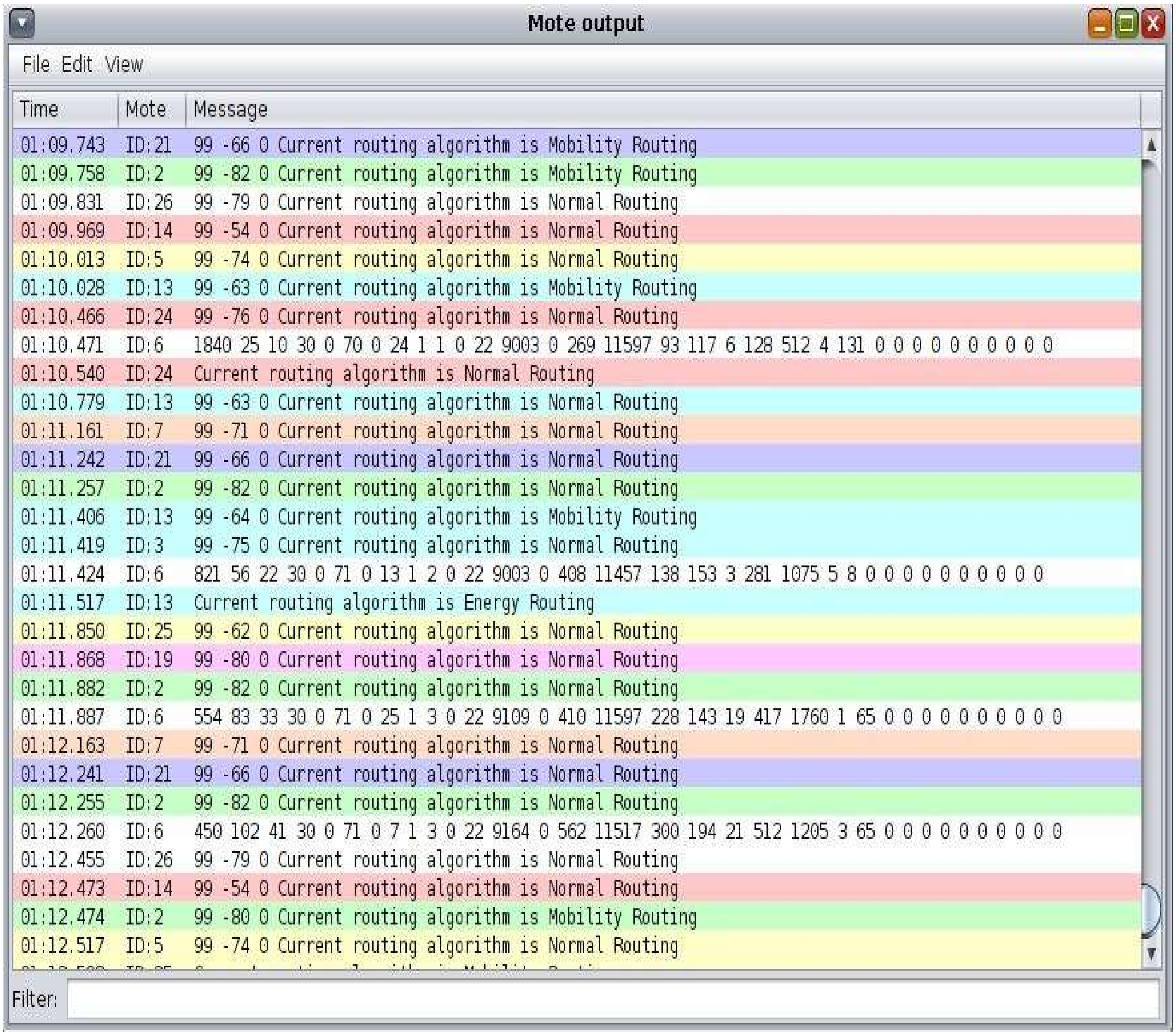
The message shows the chosen routing protocol in use according to the predefined routing thresholds, with the RSSI reading determining the mobility status of the node. Whenever a node's RSSI value changes from the previously received value, it signifies that a particular node changes its position with respect to the root node, hence mobility routing is automatically initiated. Similarly, when a node pauses, its RSSI readings remains the same with the last received RSSI value and the resultant routing protocol will be the normal routing protocol (default routing protocol).

The following RPL parameters were modified for improved performance;

- **DIO Interval Minimum:** This controls the frequency of sending DIO messages which is an important feature for reserving energy consumption and reducing traffic overhead. It influences the performance of all the proposed OF and it is displayed at *RPL_DIO_INTERVAL_MIN* (Default is 12).
- **DIO Interval Doublings:** It specifies intervals for doubling the DIO message which allows the network to minimize traffic flow and overhead, the value is specified in *RPL_DIO_INTERVAL_DOUBLINGS* (Default is 8).
- **Duty-Cycling Interval:** specifies the number of times by which a node in the DODAG accesses the transmission medium for traffic. It is set at the *NETSTACK_RDC_CHANNEL_CHECK_RATE* having a predefined value of 16 implying that the medium is accessed 16 times or more accurately indicated as 62.5ms per second.
- **Send Interval:** indicates the number of times a packet is sent to the sink, the more frequent a message is sent, the more likely it is to consume the network resources. The parameter is set at *SEND_TIME*.

From Figure 6-6, node 2 transmitted an RSSI reading of -82dBm at exactly 01:09.758 and transmitted the same RSSI value of -82dBm at 01:11.257, this means that its previous RSSI value remains the same with the current value hence it is believed that the node is stagnant now, and the current routing protocol will be the normal routing. Additionally, ID:2 did not move when it transmitted at 01:11.257 and 01:12.255, but then its RSSI value changed at 01:12.474 from -82dBm to -80dBm, therefore the mobility routing is applied. The same concept applies to all the mobile nodes in the simulation with specific emphasis to their RSSI readings except for very few cases of false positive readings which are generated by the COOJA simulation module. However, it is believed that this problem will be eliminated once the real devices are used.

This work designed the simulation in such a way that the value of each performance metric will be displayed in the output window as shown in Figure 6-6. There are 33 entries displayed on the output window form which 14 are unused by the COOJA module where all their entries are zeros used when connected to real devices.



```
01:09.743  ID:21  99 -66 0 Current routing algorithm is Mobility Routing
01:09.758  ID:2   99 -82 0 Current routing algorithm is Mobility Routing
01:09.831  ID:26  99 -79 0 Current routing algorithm is Normal Routing
01:09.969  ID:14  99 -54 0 Current routing algorithm is Normal Routing
01:10.013  ID:5   99 -74 0 Current routing algorithm is Normal Routing
01:10.028  ID:13  99 -63 0 Current routing algorithm is Mobility Routing
01:10.466  ID:24  99 -76 0 Current routing algorithm is Normal Routing
01:10.471  ID:6   1840 25 10 30 0 70 0 24 1 1 0 22 9003 0 269 11597 93 117 6 128 512 4 131 0 0 0 0 0 0 0 0 0 0
01:10.540  ID:24  Current routing algorithm is Normal Routing
01:10.779  ID:13  99 -63 0 Current routing algorithm is Normal Routing
01:11.161  ID:7   99 -71 0 Current routing algorithm is Normal Routing
01:11.242  ID:21  99 -66 0 Current routing algorithm is Normal Routing
01:11.257  ID:2   99 -82 0 Current routing algorithm is Normal Routing
01:11.406  ID:13  99 -64 0 Current routing algorithm is Mobility Routing
01:11.419  ID:3   99 -75 0 Current routing algorithm is Normal Routing
01:11.424  ID:6   821 56 22 30 0 71 0 13 1 2 0 22 9003 0 408 11457 138 153 3 281 1075 5 8 0 0 0 0 0 0 0 0 0 0
01:11.517  ID:13  Current routing algorithm is Energy Routing
01:11.850  ID:25  99 -62 0 Current routing algorithm is Normal Routing
01:11.868  ID:19  99 -80 0 Current routing algorithm is Normal Routing
01:11.882  ID:2   99 -82 0 Current routing algorithm is Normal Routing
01:11.887  ID:6   554 83 33 30 0 71 0 25 1 3 0 22 9109 0 410 11597 228 143 19 417 1760 1 65 0 0 0 0 0 0 0 0 0 0
01:12.163  ID:7   99 -71 0 Current routing algorithm is Normal Routing
01:12.241  ID:21  99 -66 0 Current routing algorithm is Normal Routing
01:12.255  ID:2   99 -82 0 Current routing algorithm is Normal Routing
01:12.260  ID:6   450 102 41 30 0 71 0 7 1 3 0 22 9164 0 562 11517 300 194 21 512 1205 3 65 0 0 0 0 0 0 0 0 0 0
01:12.455  ID:26  99 -79 0 Current routing algorithm is Normal Routing
01:12.473  ID:14  99 -54 0 Current routing algorithm is Normal Routing
01:12.474  ID:2   99 -80 0 Current routing algorithm is Mobility Routing
01:12.517  ID:5   99 -74 0 Current routing algorithm is Normal Routing
```

Figure 6- 6: Nodes Simulation Output

This work labelled the entries as follows; Time (in hours, minutes, seconds and milliseconds), packets received (the number of packets received), Delay (ms), Data Size (64B), Time Stamp, Node ID, Sequence Number, Hop Count, Message Length, Clock, Time Synchronisation (used by sub message length, presently inactive), LMP (Low Power Mode, is simply like the low power modes in phones but used by ContikiMAC to allow devices transmit and receive information while using minimum energy), Transmit Power, Receive Power, CPU Power, Listening Power, Parent ETX, Current Metric (which represents the entire value of the routing path), Duty cycle and Beacon Intervals. The message length contains sub messages which include meta information and message structure, the meta information holds the throughput, delay, data size etc, while the message structure is the main data part called sub message.

6.6.2 Simulation Results

The adaptive objective function (OR-OF) proposed in this work was implemented and tested on a simulation environment, its performance was compared with the standardised RPL OF and the proposed energy efficient and mobility OFs in this work. The performance metrics compared with includes the PDR, energy consumption, latency, parent change and network convergence time. Each of the performance metrics was tested based on the output performance while measuring the level of suitability of the objective function as well as the benefits of utilizing an OF.

Figure 6-7 shows the percentage of the PDR value which is the total number of successfully transmitted packets versus the total number of packets sent. OR-OF shows a much better performance with a high PDR of 98.23%, 95.22% and 91.67% for 10, 20 and 30 node densities respectively. The *Imin/Idoubling* indicates the maximum time by which any two successive DIO messages will be sent in a steady and stable network condition.

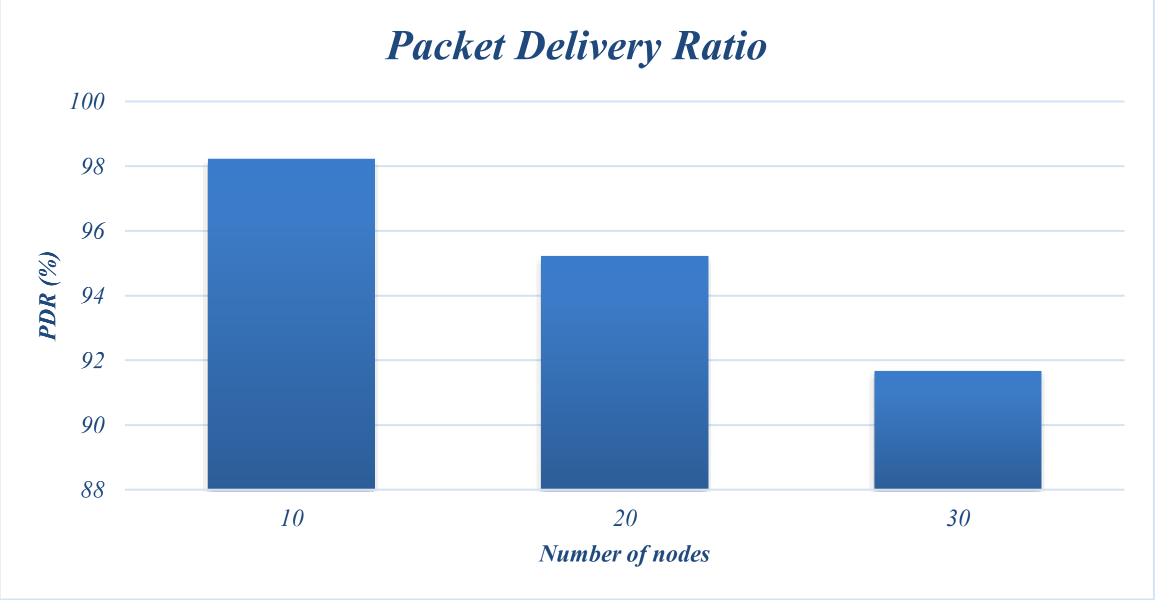


Figure 6- 7: PDR for *rpl_adaptiveof*

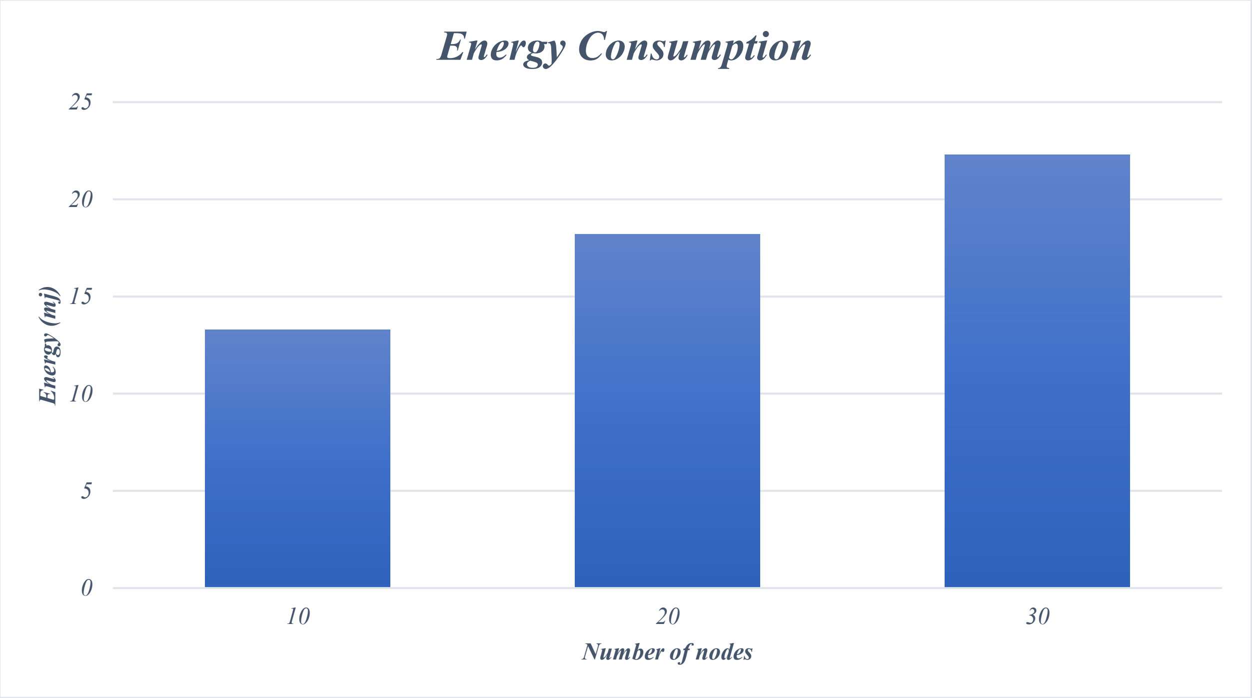


Figure 6- 8: Energy Consumption for *rpl_adaptiveof*

OR-OF is implemented in different scenarios without giving any regard for the nature of routing required at that point in time, it automatically selects the most suitable OF to be utilized using the fuzzy logic rule-based system. A PDR of 98% is the highest PDR recorded throughout the simulations with the OF utilizing both remaining energies, RSSI readings and connected energy source as thresholds for switching between the specified OF or for making routing decisions.

This approach significantly reduces the number of unwanted handovers which normally causes packet loss and congestions while indicating a high-level responsiveness towards mobile nodes. Comparing the results above with the standardised versions of RPL OF shows an overwhelming difference ergo justifying the proposal for adopting an adaptive objective function. Additionally, Figure 6-6 indicates the average energy consumption in the network for the proposed adaptive routing. It indicates the combined energy consumption for all OFs performance in the OR-OF for different node densities as compared with the standardised versions of RPL protocols. The approach proves far superior to the default versions mainly because of their inability to adapt to different routing scenarios. Similarly, the approach always selects the most optimal path at every calculation depending on the type of routing required at that point, hence it reduces the rate of retransmission and broken links that might lead to more energy consumption. The approach is highly adaptive to mobile scenarios because it encompasses the routing metrics of *rpl_mobilityof* in its OF thereby benefiting from all the advantages of enhanced mobility routing, likewise with enhanced energy saving but at the same time reducing the disadvantages of both.

Similarly, Figure 6-8 represents the average end-to-end delay of the OR-OF compared with the standardised versions of the RPL OF, this work recorded superior performance from OR-OF due to its optimal path and preferred parent selection technique. Within all the previous analysis of end-to-end delay, they were analysed together with PDR because only successfully delivered packets were considered. The adaptive routing approach shows better performance overall than the standardised versions of the RPL. From the results of the simulations conducted in the implementation of the four proposed objective functions and the standardised versions of RPL OF, this work simply presented a summary in Tables 6- 3 and 6-4 captures the percentage of increase in the analysed objective functions against some performance metrics. It specifies the exact percentage value by which the proposed objective functions outperform the standardised objective functions.

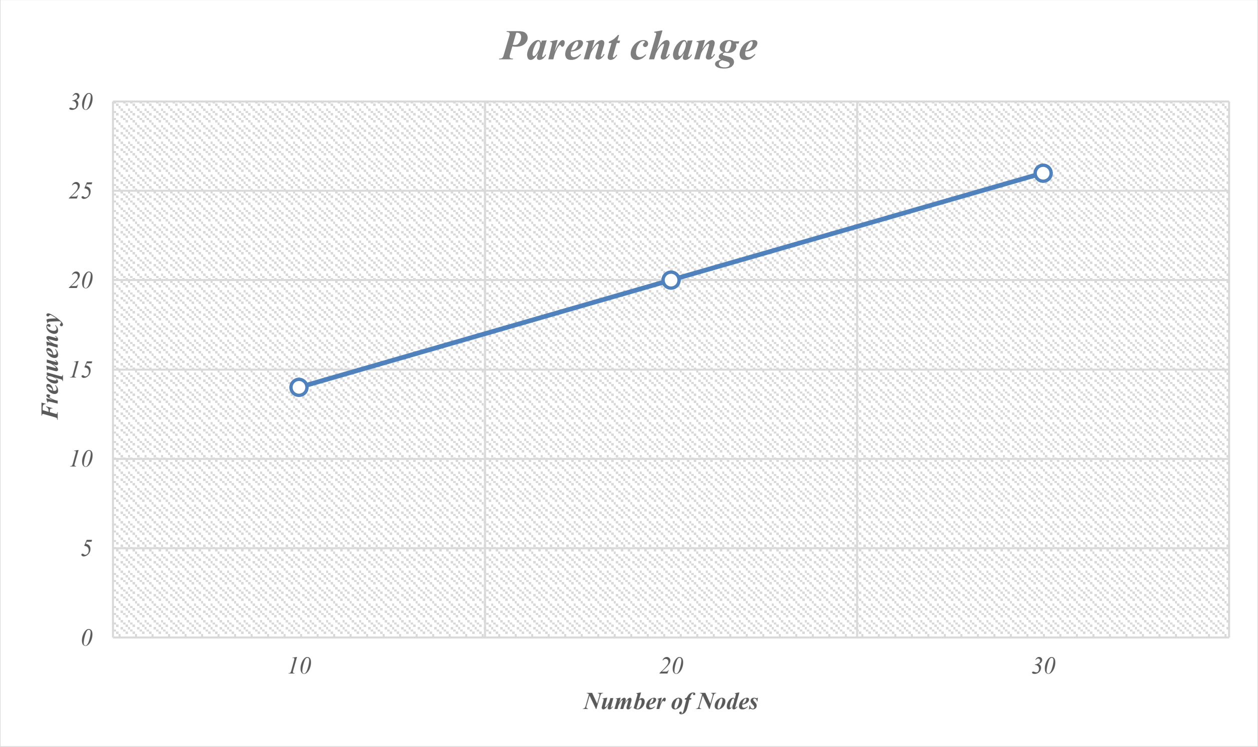


Figure 6- 9: Parent Change for *rpl_adaptiveof*

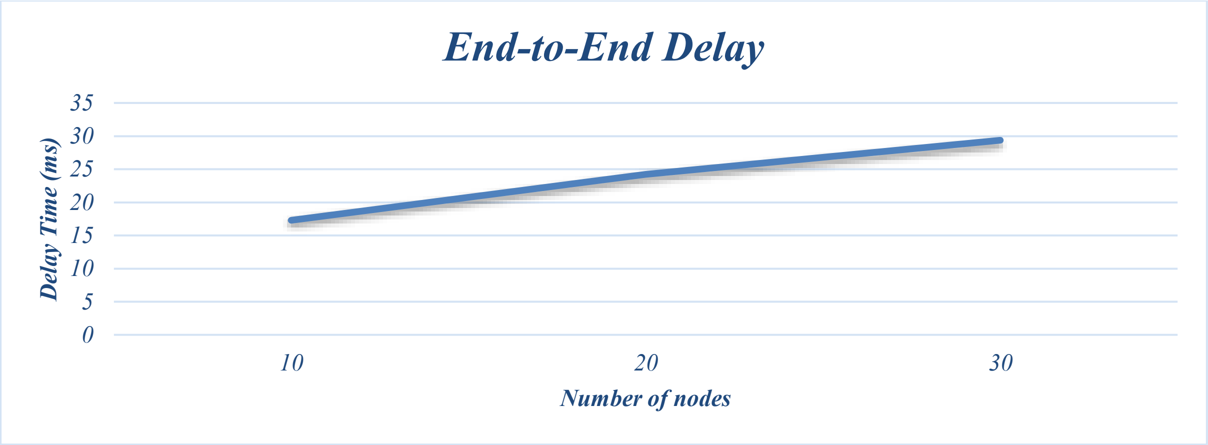


Figure 6- 10: End-to-End Delay for *rpl_adaptiveof*

Table 6- 3: Percentage Difference of Simulated Results as Compared with *rpl_OF0*

Packet Delivery Ratio									
	10 Nodes			20 Nodes			30 Nodes		
	No Mobility	RWP	RWK	No Mobility	RWP	RWK	No Mobility	RWP	RWK
rpl_energyof	15.38312%	26.76849%	21.49841%	16.17572%	25.3955%	27.18926%	17.39435%	29.80868%	28.43785%
rpl_mobilityof	6.51610%	33.46379%	31.90516%	9.666937%	31.14702%	25.85149%	13.38223%	40.1514%	29.78179%
rpl_staticof	16.72106%	-	-	19.43843%	-	-	22.16441%	-	-
rpl_adaptiveof	21.01765%	-	-	25.10644%	-	-	29.86009%	-	-
End-to-End Delay									
	10 Nodes			20 Nodes			30 Nodes		
	No Mobility	RWP	RWK	No Mobility	RWP	RWK	No Mobility	RWP	RWK
rpl_energyof	22.6997%	13.7625%	16.2359%	25.0193%	1.58535%	6.08533%	36.65160%	11.92721%	2.356685%
rpl_mobilityof	-	13.7407%	14.4075%	-	16.0289%	16.6073%	-	23.089%	24.4425%
rpl_staticof	45.5288%	-	-	41.8215%	-	-	49.59590%	-	-
rpl_adaptiveof	92.3627%	-	-	75.6706%	-	-	79.65330%	-	-
Energy Consumption									
	10 Nodes			20 Nodes			30 Nodes		
	No Mobility	RWP	RWK	No Mobility	RWP	RWK	No Mobility	RWP	RWK
rpl_energyof	35.2677%	16.6396%	10.8956%	30.9313%	55.4364%	12.5856%	37.3409%	67.3836%	30.7782%
rpl_mobilityof	-	16.7803%	13.5061%	-	50.3143%	12.70607%	-	51.1186%	0.10422%
rpl_staticof	-	-	-	-	-	-	-	-	-
rpl_adaptiveof	82.1665%	-	-	59.4627%	-	-	60.055%	-	-

Table 6- 4: Percentage Difference of Simulated Results as Compared with *rpl_mrhof*

Packet Delivery Ratio									
	10 Nodes			20 Nodes			30 Nodes		
	No Mobility	RWP	RWK	No Mobility	RWP	RWK	No Mobility	RWP	RWK
rpl_energyof	11.53094%	24.72605%	23.07388%	11.72681%	33.19333%	30.6271%	12.62134%	36.04435%	34.34239%
rpl_mobilityof	10.3789%	33.21163%	31.68855%	5.202444%	37.11217%	31.07967%	8.596693%	47.58507%	34.3157%
rpl_staticof	12.87255%	-	-	15.00091%	-	-	17.41137%	-	-
rpl_adaptiveof	17.18325%	-	-	20.69439%	-	-	25.15085%	-	-
End-to-End Delay									
	10 Nodes			20 Nodes			30 Nodes		
	No Mobility	RWP	RWK	No Mobility	RWP	RWK	No Mobility	RWP	RWK
rpl_energyof	10.0402%	19.6094%	28.4475%	12.5349%	19.4228%	14.9724%	19.4565%	2.78027%	0.67324%
rpl_mobilityof	-	28.4261%	17.7862%	-	33.6395%	25.4206%	-	37.4665%	26.0991%
rpl_staticof	33.279%	-	-	29.6283%	-	-	32.8007%	-	-
rpl_adaptiveof	82.0426%	-	-	64.6259%	-	-	64.3909%	-	-
Energy Consumption									
	10 Nodes			20 Nodes			30 Nodes		
	No Mobility	RWP	RWK	No Mobility	RWP	RWK	No Mobility	RWP	RWK
rpl_energyof	30.9625%	2.93092%	28.2629%	23.49%	38.651%	6.80883%	30.4954%	62.9507%	26.6656%
rpl_mobilityof	-	28.4017%	0.310839%	-	33.3222%	18.46128%	-	46.4541%	4.09460%
rpl_staticof	-	-	-	-	-	-	-	-	-
rpl_adaptiveof	78.4538%	-	-	52.475%	-	-	53.5757%	-	-

6.7 Numerical Analysis

In this section, this work will numerically derive and analyse the values for Packet Delivery Ratio, End-to-End delay, and Energy Consumption as utilized in this work. This section is aimed to validate simulation results with the numerical analysis of these performance metrics. It is difficult to ordinarily determine the number of packets received in a network using mathematical modelling, because there are several variables that cannot be accounted for mathematically. To this effect, this work utilized Bernoulli Probability distribution to obtain (*Psi* ψ) values, these values represent the probabilities of success in transmission.

Using Bernoulli Probability Distribution, the probability P_r that a packet is successfully transmitted across a network is given by equation 6.4 (Aburdene & Goodman, 2005):

$$P_r = \sum_{i=1}^{N} \psi^{i-1}(1-\psi) = 1 - \psi^N \qquad \mathbf{6.4}$$

Where *N* is the number of transmissions per seconds. From equation 6.4, it can be seen that when $\psi = 0$, the probability of packet transmission is equal to 1, meaning all packet transmissions are successfully received. However, when $\psi = 1$, the probability of packet transmission is equal to zero (0), meaning that none of the packets were received (total packet loss). Thus, this relationship can be explained mathematically as in equation 6.5:

$$P_r = \begin{cases} 1 & , \quad \psi = 0 \\ 0 & , \quad \psi = 1 \end{cases} \qquad \mathbf{7.5}$$

These *Psi* values in (see Appendix 3) were obtained and used to compute the PDR, end-to-end delay, and energy mathematical models. This research conducted 39 simulations in total with variations in number of nodes (10, 20, 30), node distribution and mobility model implementations. In other to validate simulation results, this work calculated the probability of success of every simulation result obtained, which will form the backbone of result validation in the mathematical models for PDR, end-to-end delay, and energy consumption. All models will be validated using values of *Psi* from (Appendix 3) in the coming sections.

6.6.1 Mobility Function

This research utilized two mobility models in the simulation which are Random Waypoint and Random Walk, but in this mathematical model this work will only focus on Random Waypoint mobility. Let $p = p_1, p_2, \dots, p_n$ represent member nodes in a network trying to send data packets to a sink node while utilizing a mobility management, with every action of the member nodes affecting the nature of communication such as change of parent node, data rate, transmission, reception and trickle settings (H. Kharrufa et al., 2018). Modelling mobility management requires deduction of the following: Nodes can send data at a fixed rate of 3 Packets/second, mobile nodes have no priorities in sending message whether they go low on energy or otherwise, nodes have mobility metric *Mm* which indicates the expected mobility intensity and with a density metric *Dm* which depends on the number of nodes, coverage, and simulation area. Similarly, this work measure node link quality *(LQ)* and *RSSI* value at time intervals with nodes utilizing the 6LoWPAN layer benefits and constraints.

The mobility function M_f introduced in this work measures the cost incurred by the existence of mobility in the network, where link cost is evaluated using *RSSI* and *LQI*. RWP mobility is used because of its replication on actual node movements within WSN (Al-Nidawi, Yahya, & Kemp, 2015) with a calculated value for *Mm*. Hence, the M_f is given by equation 6.6:

$$M_{f(p_k,\ p_{-k})} = \psi M_m LQ\lambda \qquad \mathbf{6.6}$$

Where ψ is the *Psi* value determined from Bernoulli probability distribution for mobility model, LQ is the link cost as defined by the hardware, λ is the data rate sent by nodes in the network, and M_m is the mobility metric defined in accordance with the utilized mobility scenario which can further be simplified in equation 6.7,

$$M_m = \frac{1}{|N|} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{1}{T} \int_0^T \left| V_i(t) - V_j(t) \right| dt \qquad \mathbf{6.7}$$

Where *N* is the number of node pairs, *T* is the time duration, and $V_i\,(t) - V_j\,(t)$ is the change in speed between nodes i and j at time *t*. Reflecting the actual movement of nodes using this metric in the modelling is not possible, this work intend to utilize the generated mobility scenario from the Bonnmotion as in the works of (de Waal & Gerharz, 2003).

6.6.2 Packet Delivery Ratio Model

It can be recalled from previous chapters that PDR was defined as the ratio of total number of received packets to total number of sent packets in the network which can be represented mathematically in equation 6.8:

$$PDR = \frac{R_d}{S_t} \times 100 \qquad \mathbf{6.8}$$

where R_d represents the total number of packets received, and S_t represents the total number of packets sent in the network. Furthermore, R_d can be calculated using equation 6.9:

$$R_d = P_r \times S_t \qquad \mathbf{6.9}$$

And the total number of generated packets in the network S_t can be represented mathematically in equation 6.10:

$$\boldsymbol{S_t} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{n} \times \boldsymbol{K} \times \boldsymbol{T}) \qquad \mathbf{6.10}$$

Where *T* is the duration of transmission and *n* is the number of nodes. Using equation 6.8, this work computed the PDR using the model created for this research as indicated in figure 6-11.

This research setup the model based on certain parameters, the time used in the model is 1800 seconds, with 10, 20, and 30 nodes, number of packets sent is 3 packet/second per node, although other network conditions might affect the frequency of packet transmission. Figure 6-11 shows the results of the PDR model as compared to the simulation results indicating slight difference in 10 nodes with 92.8% *rpl_energyof* and 92.32% *energy_computed* with static node setting in the network. The performance of the computed protocol did not show much difference in the PDR computation despite not including a modelling function for DIO message settings timings which has been slightly modified in the simulation setup. However, the performance of both *rpl_energyof* and *energy_computed* are better than the native standardized RPL OFs in the same simulation settings. It is believed that noise level and congestion were factors that contributed to the deterioration of packet delivery in the network. *rpl_mobilityof* is designed with the ability to adapt to certain mobility problems, it cannot proactively adapt to huge transmission rate and has shown a level of packet loss with 88.52% PDR as compared to

89.28% *Mobility_computed*, with both protocols showing reduction in PDR with increasing number of nodes.

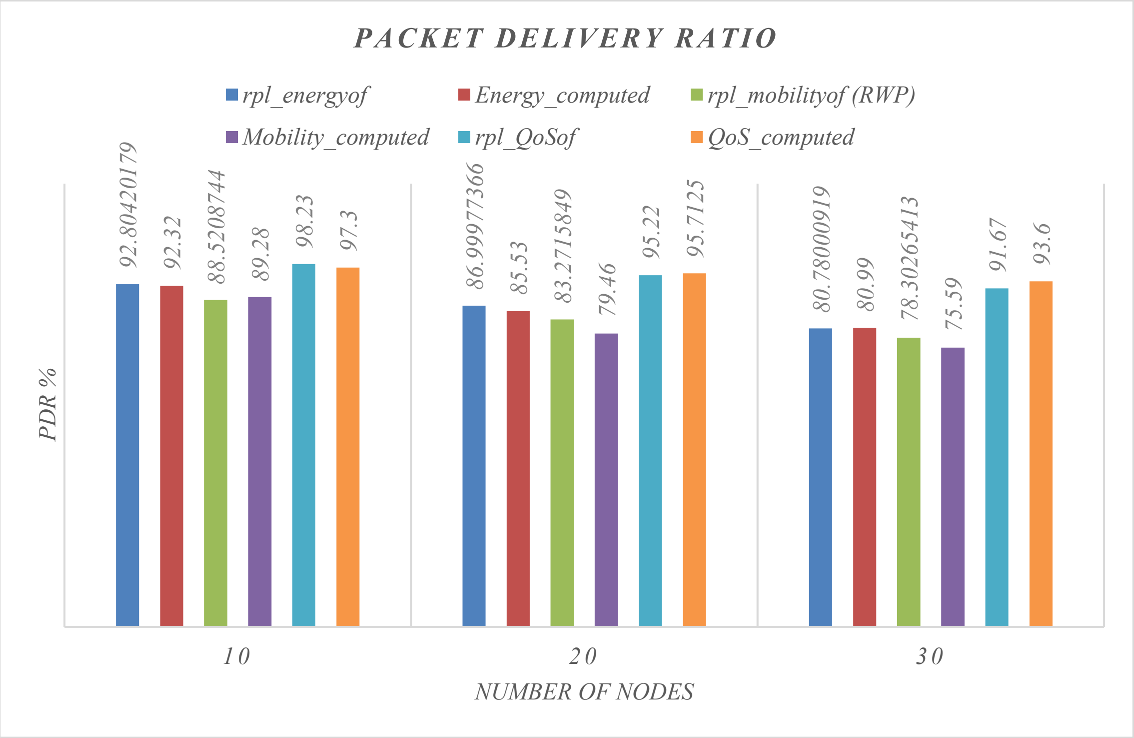


Figure 6- 11: Comparison between Computed and Modelled PDR values for different OFs

Similarly, *rpl_QoSof* and *QoS_computed* showed a PDR of 98.23% and 97.3% respectively also with a slight difference. Similarly, this work displayed the results of PDR for 20 and 30 nodes showing minimal difference in terms of performance. The reason why this work did not model the DIO, and trickle timings is that, firstly, modelling the DIO and trickle timings will produce different values for every simulation result obtained because there is no constant of proportionality in the results from the varying nodes. Secondly, there is no definite pattern that will allow the modelling to be possible, and lastly, the changes in the timings are irregular and different throughout the simulation and UDP transmission was utilized.

Generally, PDR between simulated and computed for all nodes distributions showed a slight difference in performance with little or no compensation for error correction, the simulation showed a slightly better performance because of the adjustments in trickle timer settings and DIO time slices which handles the frequency of connection and reconnection in the network. In terms of mobility, it might be observed that the PDR is slightly high, this is because mobile nodes move at same speed and sensors only move out of coverage areas for a short period. Routing overhead did not have much effect on the performance of the network simulation only slightly in mobile nodes when re-joining the network after leaving network range.

6.6.3 End-to-End Delay Model

We know that in a typical packet switching network paradigm, transmitted data packets are broken down into smaller segments from one endpoint of the network to another, which is then later packetized at its destination. This process involves certain steps as the data segments are transmitted from hosts, access points and routers, where intermediate nodes receive data and forward them by placing them in queues at the buffer. The buffer size depends on the network transmission card of the devices in use. This work will closely look at the different types of delay in the network as discussed in (Boris B., 2015):

Transmission Delay: denoted by D_s is known as the amount of time needed to transmit a packet at a specific transmission rate. If *L* is packet length in bits and *R* is transmission rate in bps, then transmission delay is given in equation 6.11:

$$D_s = \frac{L}{R} \qquad \mathbf{6.11}$$

Propagation Delay: denoted by D_p be the amount of time needed for a packet to travel through the network from source to destination which depends on the distance and medium characteristics, where *d* is the distance of receiver in meters and *v* is the medium characteristics used in communication which can be wireless network (3 x 10^8 m/s), fibre optic cable (2 x 10^8m/s), etc, then the propagation delay is given in equation 6.12:

$$D_p = \frac{d}{v} \qquad \mathbf{6.12}$$

Queueing Delay/Waiting Time: denoted by D_q represents the time spent by a packet in a queue waiting for onward transmission. This aspect is the study interest in the coming equations because it depends on the amount of network traffic caused by packets and cannot easily be determined by basic parameters.

Processing Delay: denoted by D_c represents the time required for a packet to be analysed by the receiver, which includes error checking or size of routing table completely hardware specific.

Total Delay in a Node: denoted by D_{node} represents the packets overall time in a node which is the sum of transmission, processing, and propagation delays. Hence it can be represented mathematically as in equation 6.13:

$$D_{node} = D_s + D_c + D_p \qquad \mathbf{6.13}$$

Total Delay in a Hop: denoted by D_{hop} represents the time spent by a packet in a hop, which is given mathematically as in equation 6.14:

$$D_{hop} = D_{node} + D_q \qquad \mathbf{6.14}$$

End-to-End Delay: denoted by D_{e2e} represents the total time spent by a packet traversing from the source to the receiver, which depends on the hops and its characteristics (traffic load, distance, etc.). it can be shown mathematically as in equation 7.15:

$$D_{e2e} = \sum_{\forall i} D_{hop,i} \qquad \mathbf{7.15}$$

Most of the delays described above are somewhat static, that is they do not change over a period of transmission (transmission, propagation, and mostly processing). The dominant variance of end-to-end delay in constant flow packet network is queueing delay which will be the focus of analysis in this work. Given that $N(t)$ is the number of packets over time, which is related to packet arrival and departure functions as $\lambda \; and \; \mu$ respectively, this work can easily derive the exact value of end-to-end delay in the system (Boris B., 2015). Although transmission delay can always remain the same given that all packets have the same size, but queueing/waiting delay always vary as it depends on the buffer state. The building blocks of the analytical model is given in table 6-5.

The service transmission time denoted by $E[D_s] = \frac{E[L]}{R}$, while the maximum packet departure rate is given by $S\mu = \frac{S}{E[D_s]}$, with traffic intensity $\alpha = \frac{\lambda}{\mu}$. this work will be using the Erlang (carrying load capacity of telephone networks) notation for network model formation of

M/M/S/K and *M/M/1*, where the first and the second *M* represents Poisson arrival process and exponentially distributed service times respectively, *S* denotes the server and *K* the buffer capacity (packets in both buffer and service, if omitted, it is assumed that buffer is infinite).

Table 6- 5: Definition of Parameters for End-to-End Delay

Value	*Definition*
$\boldsymbol{S}$	*Number of servers (i.e. transmitters or processors)*
$\boldsymbol{K}$	*Packets in the System*
$\boldsymbol{Q}$	*K – S (packets in the buffer)*
$\boldsymbol{R}$	*Transmission rate (bps)*
$\boldsymbol{\lambda}$	*Aggregate packet arrival rate (packets/second)*
$\boldsymbol{L}$	*Packet Length*
$\boldsymbol{\mu}$	*Packet departure rate*
$\boldsymbol{\alpha}$	*Traffic intensity*
$\boldsymbol{P_b}$	*Packet loss probability*
$\boldsymbol{\rho}$	*System utilization*
$\boldsymbol{\pi}$	*Poisson arrival process*
$\boldsymbol{E[N]}$	*Expected number of packets in the system*
$\boldsymbol{E[N_s]}$	*Expected number of packets in service*
$\boldsymbol{E[D_s]}$	*Expected delay of packets in the server*
$\boldsymbol{E[D_q]}$	*Expected delay of packets in the buffer*
$\boldsymbol{E[D]}$	*Expected delay of packet in the system*
$\boldsymbol{E[N_q]}$	*Expected number of packets in the queue*

Where service rate completely depends on the state of the system implying that busy server means higher service rate, the relationship between service rate and system state can be depicted in Figure 6-12.

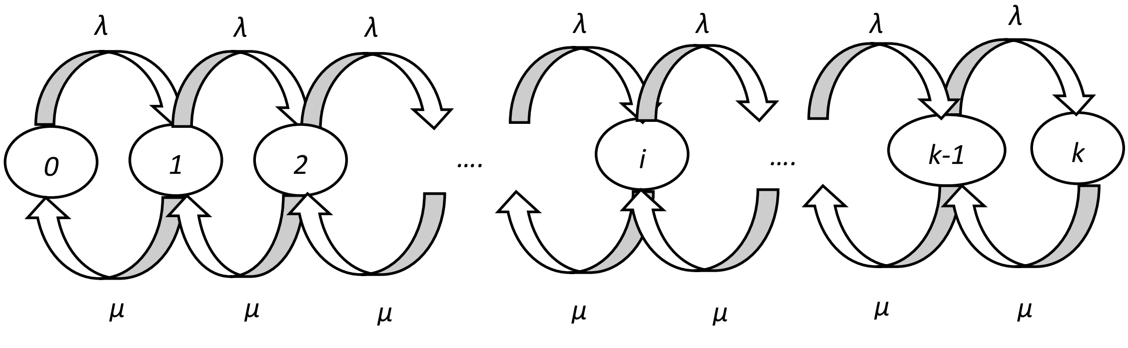


Figure 6- 12: Markov Chain for Queue Delay of M/M/1 (Boris B., 2015)

Figure 6-12 can be expressed in terms of a load balancing equation as follows

$$\pi_0\lambda = \pi_1\mu$$

$$\pi_1\lambda = \pi_2\mu$$

$$\dots$$

$$\pi_{i-1}\lambda = \pi_i\mu \rightarrow \pi_i = \left(\frac{\lambda}{\mu}\right)\pi_{i-1} = \alpha\pi_{i-1} \rightarrow \pi_i = \alpha^i\pi_0$$

$$\dots$$

$$\pi_{k-1}\lambda = \pi_k\mu$$

Some other parameters utilized are defined as follows;

- The probability that a buffer will be empty at a given time is $P_e = \pi_0$
- The probability that there are *i* packets at any given time is π_i
- The probability of packet loss is $P_b = \pi_k$
- The fraction of time by which the system will be actively transmitting packets or system utilization is $\rho = 1 - \pi_0$

Using a normalization condition to obtain the equilibrium distribution where the sum of all state probabilities is equal to one, and the initial state (*0th* state) probability for *M/M/1/K* is given by equation 6.16:

$$\pi_0 = \frac{1}{\sum_{j=0}^{k}\alpha^j} = \frac{1}{\frac{1-\alpha^{k+1}}{1-\alpha}} = \frac{1-\alpha}{1-\alpha^{k+1}} \qquad \mathbf{6.16}$$

However, the equilibrium distribution for *M/M/1* queue system is not far off from the *M/M/1/K* system when $K \rightarrow \infty$, which will make $\propto^{k+1} = 0, if\ \alpha < 1$ and *0th* state probability given by equation 6.17:

$$\pi_0 = \frac{1}{\sum_{j=0}^{k}\alpha^j} = \frac{1}{\frac{1}{1-\alpha}} = 1 - \propto \qquad \mathbf{6.17}$$

While the *0th* state probability for both queueing systems is known, the *ith* probability is given by equation 6.18 and equation 6.19 respectively,

$$\pi_i = \alpha^i\pi_i = \frac{(1-\alpha)\alpha^i}{1-\alpha^{k+1}} \qquad \mathbf{6.18}$$

$$\pi_i = \alpha^i \pi_0 = (1-\alpha)\alpha^i \qquad \mathbf{6.19}$$

Similarly, the probability of packet loss due to overflow is equal to probability that a packet meets a full system or k^{th} state for both systems can be expressed in equations 6.20 and 6.21:

$$P_b = \pi_k = \frac{(1-\alpha)\alpha^k}{1-\alpha^{k+1}} \qquad \mathbf{6.20}$$

$$P_b = 0 \qquad \mathbf{7.21}$$

In other to know the number of packets in the system, the average number of packets in queue (queue occupancy) as long as $\alpha \neq 1$ *and* $\alpha < 1$ will be calculated, if that is the case then the following equations 6.22 and 6.23 holds;

$$E[N] = \sum_{q=0}^{k} \pi_q q = \frac{\alpha}{1-\alpha} - \frac{(k+1)\alpha^{k+1}}{1-\alpha^{k+1}} \qquad \mathbf{6.22}$$

$$E[N] = \sum_{q=0}^{\infty} \pi_q q = \frac{\alpha}{1-\alpha} \qquad \mathbf{7.23}$$

While for α = 1 meaning no packets in the buffer, which implies that there is no point of computation. The work is based on the fact that at every time a packet is received it meets another packet in the buffer. This work utilized Poisson arrival process and Little's law in the analysis of *M/M/S/K and M/M/1* queue system, where the average number of packets in the system is obtained by simply multiplying the average time the packets spent in the system by the arrival rate, which can be represented from equations 6.24 to 6.29:

$$E[D] = \frac{E[N]}{\lambda(1-P_b)} \qquad \mathbf{6.24}$$

$$E[D_q] = \frac{E[N_q]}{\lambda(1-P_b)} \qquad \mathbf{6.25}$$

$$E[D_s] = \frac{E[N_s]}{\lambda(1 - P_b)} \qquad \mathbf{6.26}$$

$$E[N_q] = \sum_{i=S+1}^{K} (i - S) * \pi_i = E[N] - E[N_s] \qquad \mathbf{6.27}$$

$$E[N_s] = \sum_{i=0}^{K} min(i, S) * \pi_i \qquad \mathbf{6.28}$$

$$E[N] = \sum_{i=0}^{K} i * \pi_i \qquad \mathbf{6.29}$$

As stated earlier, the above network model followed the *M/M/S/K* queuing system of the Markov chain which implicitly assumes the Poisson arrivals and exponentially distributed service times. Additionally, the *M/M/1* queuing system provides a complementary tool of analysis which is often utilized if certain conditions hold, firstly, the value of *K* must be infinite meaning that it approaches infinity which requires the value $\alpha < 1$, secondly, the queue size must be large and lastly the traffic intensity must be low. These prerequisites provide a very accurate model and allows for different delay estimations than the *M/M/1/K* queue using simpler expressions. The system occupation for *M/M/1* can be expressed as in equations 6.30 to 6.35:

$$E[N] = \sum_{q=0}^{\infty} \pi_q q = \frac{\alpha}{1 - \alpha} \qquad \mathbf{6.30}$$

$$E[N_s] = 1 - \pi_0 = \alpha \qquad \mathbf{6.31}$$

$$E[N_q] = E[N] - E[N_s] = \frac{\alpha}{1 - \alpha} - \alpha = \frac{\alpha^2}{1 - \alpha} \qquad \mathbf{6.32}$$

$$E[D_s] = \frac{E[N_s]}{\lambda(1 - P_b)} = \frac{\alpha}{\lambda} = \frac{1}{\mu} \qquad \mathbf{6.33}$$

$$E[D] = \frac{E[N]}{\lambda(1 - P_b)} = \frac{1}{\mu(1 - \alpha)} = \frac{1}{\mu - \lambda} \qquad \mathbf{6.34}$$

$$E\left[D_q\right] = \frac{E\left[N_q\right]}{\lambda(1 - P_b)} = \frac{\alpha^2}{\lambda(1 - \alpha)} = \frac{\alpha}{\mu(1 - \alpha)} = \frac{\alpha}{\mu - \lambda} \qquad \mathbf{6.35}$$

All the above assumptions made followed the Burkes and Jackson's theorems (Tsitsiashvili & Osipova, 2018) who proved that the *M/M/S* queue system follows a Poisson distribution process that allows us to model independent behaviours of network interfaces.

Calculating an end-to-end delay in a system is the summation of the average time a packet spends in each hop and the average delay of packets in the system which is represented in equations 6.36:

$$D_{e2e} = \sum_{\forall i} \left(E[D_{node}] + E\left[D_q\right]\right)$$

$$= \frac{1}{\mu - \lambda} + \frac{\alpha}{\mu - \lambda}$$

$$D_{e2e} = \sum_{\forall i} \frac{1 + \alpha}{\mu - \lambda} \qquad \mathbf{6.36}$$

This work further incorporated number of nodes, hop count and time into the overall end to end delay into the model, which translates into equation 6.37:

$$D_{overall} = \psi D_{e2e}\, H_i\, n\, t \qquad \mathbf{6.37}$$

$$H_i = H_{i-1} + \frac{1}{8} H_0, \qquad i \geq 2, \qquad \mathbf{6.38}$$

Where H_0 is the hop count which has been initialised to a minimum of 2, *n* is the number of nodes, and *t* is the duration time. In the arrangement of nodes for the model, nodes have a maximum of two hops except where nodes are mobile. The parameters utilized for calculating the model is given in table 6-6.

Table 6- 6: Numerical Values for Calculating End-to-End Delay

Variable	*Value*	***Unit***
$\boldsymbol{\lambda}$	(30, 60, 90) * P_r	***Packets/Seconds***
Packet Size	*64*	***Bytes***
$\boldsymbol{E[L]}$	*512*	***Bits***
$\boldsymbol{R}$	*50*	***Kbps***
$\boldsymbol{E[D_s]}$	*0.00128*	***Seconds***
$\boldsymbol{\mu}$	*800*	***Packets/Seconds***
$\boldsymbol{\alpha}$	$0 < \alpha < 1$	***Erlangs***
$\boldsymbol{n}$	*10, 20, 30*	-
$\boldsymbol{t}$	*1800*	***Seconds***
$\boldsymbol{H_0}$	*2*	-

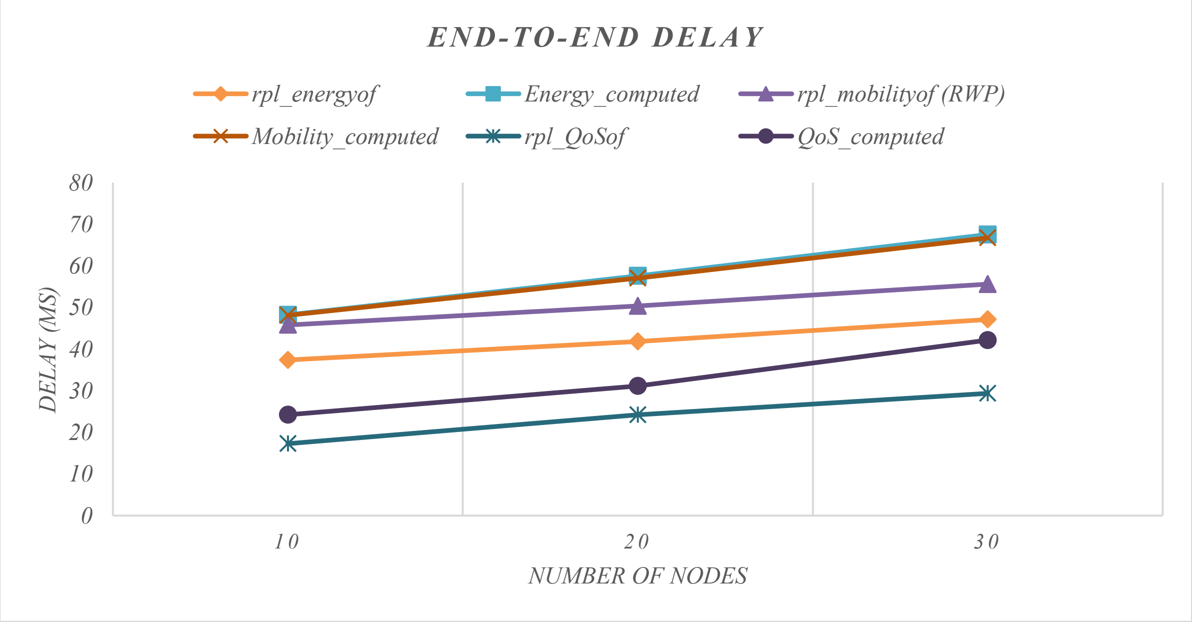


Figure 6- 13: Comparison between simulated and Modelled results between end-to-end delay values for different OFs.

The average computed delay as compared to the simulated results is given in Figure 6-13, which gives the delay time taken for packet to travel from the application layer of the sending node to the application layer of the receiving node. This work utilized a transmission rate of 3 packets/second where the entire simulation and computed results do not show huge difference in terms of performance results due to the low number of nodes, and rate of transmission did not affect the link quality cost. At 3 packet/second transmission rate in the network, *energy_computed* shows a slightly higher result with about 11, 16 and 20 ms for 10, 20 and 30 nodes respectively as compared to *rpl_energyof*, while *mobility_computed* shows a difference of 3, 7 and 11 ms for 10, 20 and 30 nodes respectively as compared to *rpl_mobilityof*, and *QoS_computed* shows a difference of 6, 7 and 13 ms respectively for 10, 20 and 30 nodes as compared to *rpl_QoSof*. High delay is attributed to network congestion and high transmission rate of packets which also increases in the presence of mobility. Additionally, this work believes that the difference between the simulated and computed results is due the adjustment in DIO and trickle timer settings as discussed in previous chapters which will compensate for the error correction.

6.6.4 Energy Consumption Model

Reflecting the true energy consumption using the derived energy consumption model shows the main energy constraints in Contiki IoT environment can be achieved by explaining the following equations in detail. The network stack of the Contiki utilizes the Radio Duty Cycling (RDC) mechanism which allows the device radio to remain active and listen to transmissions for certain periods in time, as for LPM mode no reception of transmitted packets. It is a form of mechanism which significantly reduces energy consumption in transmission where the radio remains on listening mode which can either be synchronous or asynchronous. Synchronous uses the TDMA mechanism that allows devices to wake up at beginning of reception for incoming packets for fixed period, while asynchronous only ensures reception of packets destined for it. Overall energy consumption based on the duty cycle of a device can be expressed in terms of Transmission *Tx*, Reception *Rx, LPM* and *CPU_time* which are explained in equation 6.39 (Lekidis & Katsaros, 2018; R. K. Singh, Puluckul, Berkvens, & Weyn, 2020):

$$E_m = \left(\sum_{\forall i \in D_{LPM}}^{N_{LPM}} I_{LPM} * V_{LPM} + \sum_{\forall j \in D_{Tx}}^{N_{Tx}} I_{Tx} * V_{Tx} + \sum_{\forall k \in D_{Rx}}^{N_{Rx}} I_{Rx} * V_{Rx} + \sum_{\forall z \in D_{CPU}}^{N_{CPU}} I_{CPU} * V_{CPU} \right) * \Delta t \qquad \mathbf{6.39}$$

where *I, V* and Δt indicate the current (Ampere), voltage (Volts) and time intervals between operating modes, and $N_{LPM}, N_{Tx}, N_{Rx} \ and \ N_{CPU}$ represents the frequency of device visiting different operating modes and E_m represents the energy model. Equation 6.39 can further be represented as equation 6.40:

$$E_m = \left(\frac{V \, \Delta t \sum_{i=1}^{4} I_i}{B_{tt}}\right) n \qquad \mathbf{6.40}$$

Where B_{tt} represents the battery capacity and I_i represents the different currents for transmission, reception, *CPU_time* and *LPM* modes which are given by *19.5mA*, *21.5mA*, 1*.8 mA* and *0.0545 mA* respectively. The voltage is given as *3V* for IoT systems transmission which is the same for all transmission modes and B_{tt} represents the total energy in milli-joules *(mj)* which is given by *32768* (Lamaazi & Benamar, 2018). This work modified equation 7.40 to produce a generic energy consumption model for the numerical analysis which includes the number of nodes, probability of success and energy consumed by peripheral devices as given in equation 6.41:

$$E_{ci} = \psi E_m \, E_t^i \qquad \mathbf{6.41}$$

Where ψ represents *Psi* probability of success as defined in Table 7-5, with *i=1,2,3, ..., q* and E_t represents the energy consumed by peripheral devices. It is a fact that within the peripheral device itself, there is an amount of energy consumption from the embedded system which comes from peripheral devices such as video, audio or network devices (Celebican, Rosing, & Mooney III, 2004). Modelling the peripheral energy consumption of nodes is not within the scope of this work, but energy consumption value for devices will be adopted from previous research works. However, E_t^i can further be broken down into equation 6.42,

$$E_t^i = \begin{cases} E_t^1 \,, & 10 \, nodes \\ E_t^2 \,, & 20 \, nodes \\ E_t^3 \,, & 30 \, nodes \end{cases} \qquad \mathbf{6.42}$$

Where *i = 1,2,3.* Equation 6.42 is a step function which shows different levels of peripheral device energy consumption of 10, 20 and 30 nodes as utilized in the work. Similarly, this is a generic model which can be used in any number of nodes if properly defined.

If we define I_o as the initial energy before transmission, hence the remaining energy RM_{energy} of a node is given by equation 6.43:

$$RM_{energy} = I_o - E_{ci} \qquad \mathbf{6.43}$$

Since this work will be using more than one metric for path and preferred parent computation, with each of the metrics having its distinct unit for computation, it must unify the metrics into a single value (Abubakar Gidado Halilu, M. Hope, Haifa Takruri, & U. Aliyu, 2020).

Figure 6-14 shows computed energy consumption as compared to simulated results displaying different levels of consumption. At 3 packet/second transmission rate, *energy_computed* shows a little higher consumption with 0.5, 8 and 8.4mj for 10, 20 and 30 nodes respectively as compared to *rpl_energyof*.

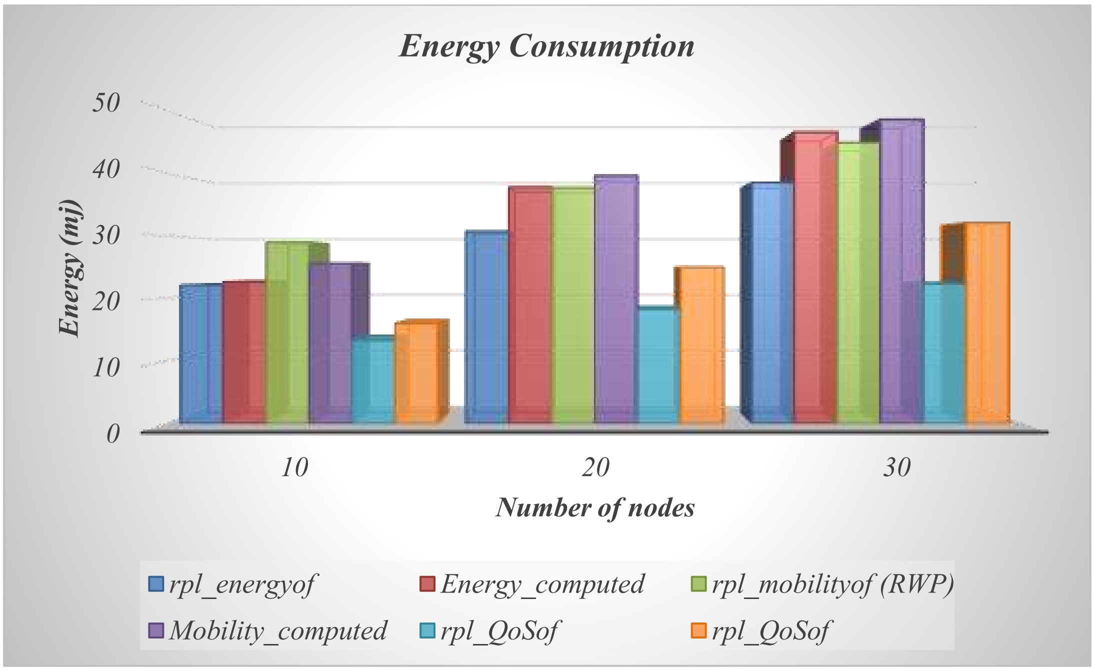


Figure 6- 14: Comparison between Computed and Modelled energy consumption values for different OFs

Similarly, *mobility_computed* shows certain differences in performance with 3mj lower, 2 and 4mj higher for 10, 20 and 30 nodes respectively as compared to *rpl_mobilityof*, and *QoS_computed* shows higher difference in 10, 20 and 30 nodes with 3, 7 and 10mj respectively as compared to *rpl_QoSof*. The same effect in the case of PDR and end-to-end delay also applies to energy consumption in terms of DIO and trickle timer settings in controlling network overhead.

6.8 Comparing Research Results with other Related Works

In this section, the research will aim to compare results both simulated and computed results with other related studies in this area. Results will indicate difference in performance which are often inclined towards difference in design, data set, performance metric comparisons, and other factors. The work to compare is done by (H. D. Y. Kharrufa, 2018) which has a similar approach by firstly conducting simulation and then building a mathematical model to validate simulation results.

The simulation results for PDR in (H. D. Y. Kharrufa, 2018) has been done in several areas like healthcare and animal tracking with values of 84% and 78% respectively respective for 25 node distribution, while this research achieved 87% and 81% in *rpl_energyof* for 20 and 30 node distribution respectively. This research further achieved 72% and 68% in *rpl_mobilityof* for 20 and 30 node distribution respectively, both results for PDR are displayed in Figure 6-15.

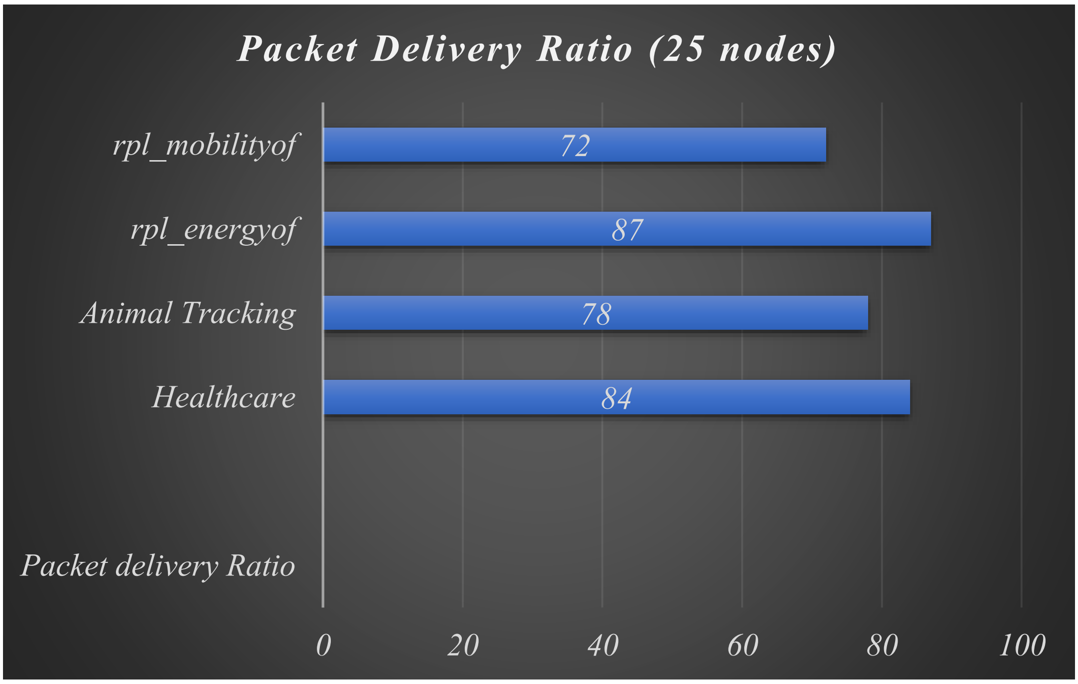


Figure 6- 15: Comparison of PDR between (H. D. Y. Kharrufa, 2018), *rpl_energyof* and *rpl_mobilityof.*

Similarly, (H. D. Y. Kharrufa, 2018) obtained average delay of 200ms for animal tracking while this research obtained 53ms and 59ms in *rpl_mobilityof* for 20 and 30 node distribution respectively, as displayed in Figure 6-16.

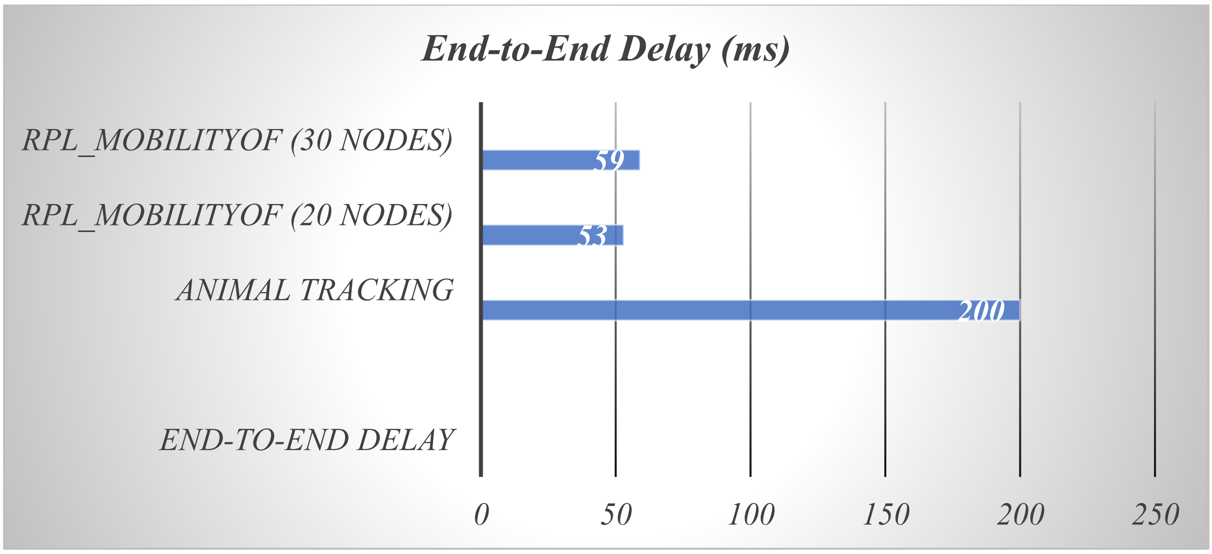


Figure 6-16: Comparison of End-to-End delay between (H. D. Y. Kharrufa, 2018), *rpl_energyof* and *rpl_mobilityof*.

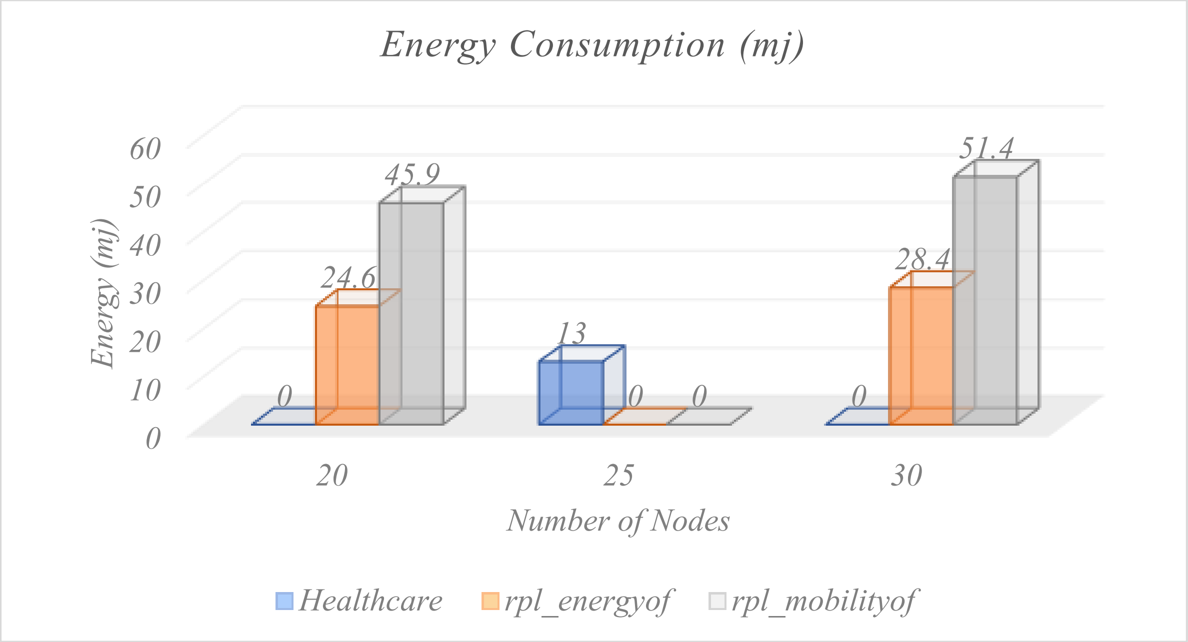


Figure 6-17: Comparison of Energy Consumption between (H. D. Y. Kharrufa, 2018), *rpl_energyof* and *rpl_mobilityof*.

The work compared with has shown a very high delay due the concept of modifying trickle timer algorithm which causes a very high computational and processing overhead. This however did not significantly affect the PDR because they increased the size of their buffer occupancy, hence more packets are often stored in the buffer which prevents packet loss and congestion. Furthermore, the compared research also evaluated their energy consumption model on medical healthcare with an average of 13mj for 25 node distribution, while this research recorded 24.6mj and 28.4mj in *rpl_energyof* for 20 and 30 node distribution respectively as shown in Figure 6-17. The major reasons for differences between the two research are mainly based on the design consideration, data set, testing environment, researcher perspective/interest, and modification criteria.

The second aspect of this comparison is the mathematical modelling which was used to validate simulation results. (H. D. Y. Kharrufa, 2018) obtained a computed PDR of 90% with 12mj energy consumption and 270ms end-to-end delay, while this research obtained PDR of 92%, energy consumption of 22.6mj, and end-to-end delay of 48ms for *rpl_energyof*, as well as PDR of 89%, energy consumption of 25.3mj, and end-to-end delay of 48ms for *rpl_mobilityof* as displayed in Figure 6-18.

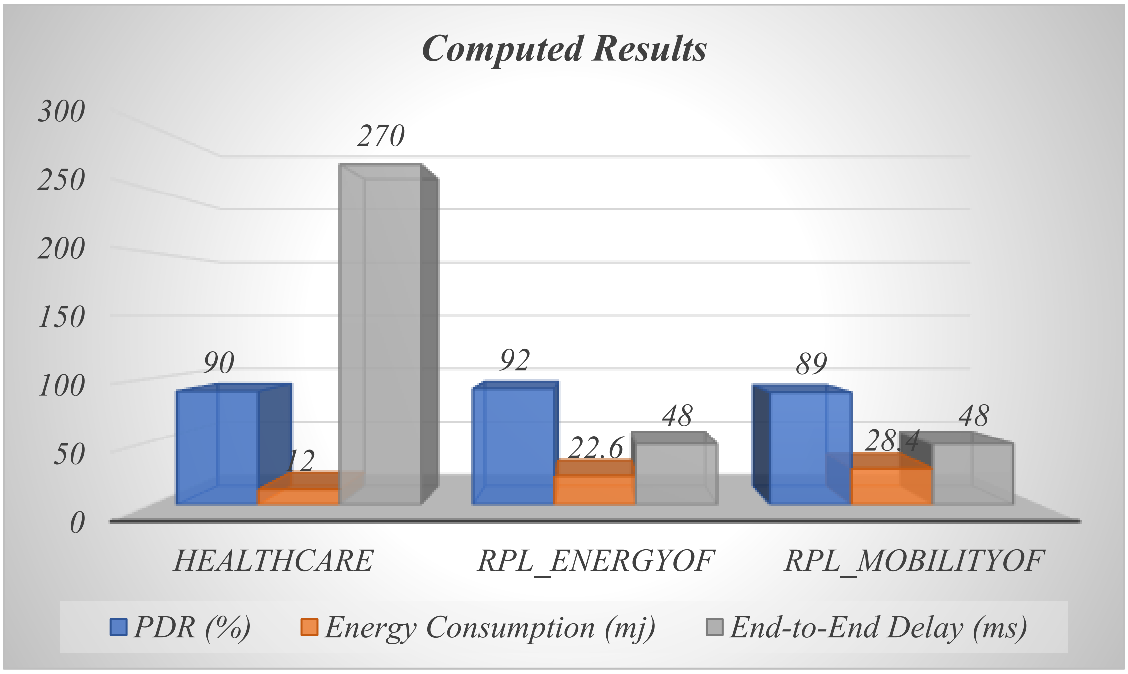


Figure 6-18: Comparison of Computed results between (H. D. Y. Kharrufa, 2018), *rpl_energyof* and *rpl_mobilityof*.

6.9 Summary

This work presented two comprehensive Tables in 6-3 and 6-4 which captures all the percentage difference in performance of the proposed simulated routing protocols as compared with the standardised versions of RPL OF. The simulated protocols showed superior performance in all aspects of routing, with implementation and discussion on the performance evaluation of *rpl_adaptiveof* as compared with standardised versions of RPL OF for adaptive routing in both dynamic and static IoT applications in handling their different needs by providing highly robust, reliable, scalable, and efficient algorithm. Generally, *rpl_adaptiveof* showed a better performance overall in terms of latency, packet delivery, energy consumption and parent change, which were implemented using two mobility models with varying node densities as simulated in Chapter 4 of this work. the approach achieves an overall average of 98% PDR, 17.29ms latency and 13.30mj energy consumption as compared to the standardised version of RPL OF. This approach is the backbone of this research as it combines all the developed OFs into a single OF and simulation results indicates significant improvement on the entire performance of RPL protocol when handling different scenarios. The approach relies heavily on three metrics remaining node energy, RSSI reading and Power source which are the major metrics consideration in adaptive routing, the remaining metrics are latency, hop-count, ETX and Link quality which were used for path and preferred parent selection.

Additionally, this work developed mathematical models for PDR, end-to-end delay, and energy consumption in the work, then compared the performance of the simulated protocols with the modelled results and recorded the differences. Simulated results are slightly better than modelled results due to the adjustments of DIO message timings and trickle timer settings which influenced the performance.

Chapter Seven

Conclusion and Future work

7.1 Conclusion

This research proposed several objective functions (OF) that provided improved performance of RPL routing protocol in terms of energy efficiency, mobility management and support for energy scavenging nodes using simulations of different antagonistic application requirements. The aim of this research which was capped in three phases which were all achieved; firstly, we provided a complete simulation set of the proposed objective functions and implement them on Cooja Contiki operating system where we recorded numerous improvements as compared to the standardized system. Secondly, we developed a mathematical model for the evaluation and validation of simulation results. The adaptive objective function is set to provide the growing needs of future IoT applications. This research achieved all our set out objectives by identifying and filling research gaps, using key design tools and parameters, improvement on OF algorithms and validation with suitable mathematical models. This research has recorded more than 50% overall performance improvement over the standardized versions as summarised in Table 6-3 and 6-4 of this research on different performance metrics such as PDR, energy consumption, and end-to-end delay.

This chapter gives the research output by outlining the major contributions and findings in each chapter. It further explains the relevance of the novelties in ensuring enhanced energy efficiency and mobility in IoT applications. In chapter 4, an enhanced mobility support routing protocol for Low Power and Lossy Networks for IoT applications was introduced, the proposed routing protocol modifies the objective function to include additional metrics in terms of path and preferred parent selection. It improved the availability, parent selection and network lifetime of nodes in the network with at least 45% compared to the default objective functions. Simulation results indicate that *rpl_mobilityof* presented a satisfactory level of mobility management, improved energy consumption and reduced end-to-end delay with high response towards path and preferred parent changes in the network. It utilizes the RSSI readings to determine the mobility state of each node and the OF was simulated with 10, 20 and 30 nodes implemented with Random WayPoint and Random Walk Mobility models, and recorded an average packet delivery ratio of 88%, end-to-end delay of 45.7ms and an energy consumption of 28.90mj as compared to the standardised versions of RPL OF. The packet delivery ratio is not 100% which indicates the likelihood of packet drop and QoS was not guaranteed in this OF.

In chapter 5, an enhanced energy efficient routing protocol for IoT applications was introduced (*rpl_energyof*) and it was implemented by combining the remaining energy, link quality and estimated transmission count within the OF for path and preferred parent selection. Simulation results indicates superior energy saving capabilities with over 50% increase over the standardized version, improved network lifetime, improved network convergence time and a satisfactory level of parent change. This work recorded average values of 92.80% packet delivery ratio, 37.39ms of end-to-end delay and 22.30mj of energy consumption as compared to the standardised versions of the RPL OF. It shows a significant reduction of energy consumption of about 20% as compared to the proposed objective functions and the standardised versions of the objective function. Energy efficiency is an important consideration in routing as it is one of the major challenges of a routing protocol, hence most research in routing is always geared towards improving node energy consumption for improved network lifetime. This work proposed an optimized routing protocol for energy scavenging nodes where the research combined the standardised objective functions into a single OF used for static nodes. This proposed energy scavenging objective function takes into consideration all those nodes that are either mains-powered or have an alternate energy source e.g., solar, with combination of both node and link metrics. This work combined ETX and hop-count metric as a single OF known as *rpl_QoSof* and simulation results indicate a packet delivery ratio of 94.06% and an end-to-end delay of 29ms as compared to the standardised versions of RPL OF in a static setting. This work assumed that nodes are carefully placed within the transmission range of one another, and all nodes have an energy source that is not battery-powered, hence this setting can guarantee a high QoS.

In chapter 6, this work proposed an adaptive objective function in RPL protocol known as OR-OF which combines the three proposed OF and the default standardised OF to perform as a single OF that can suit different IoT application requirements. This work utilized the fuzzy membership function with the fuzzy logic system to determine the most suitable paths or preferred parents to the destination. This work utilized six input metrics of both node and link which passes through the knowledge-based and inference engine of the fuzzy logic system to select the appropriate routing protocol from *rpl_mobilityof, rpl_energyof, rpl_QoSof* and *rpl_mrhof*. Simulation results indicate a superior performance on all the previously proposed OFs in this research, recorded a whopping 98% packet delivery ratio, with 17.29ms end-to-end delay and 13.30mj of energy consumption. It collectively considers static nodes, mobile nodes, energy scavenging nodes, grid and randomly positioned nodes, it ensures that the best routing situation is selected at any given time. The adaptive approach improves throughput, availability, end-to-end delay, PDR, and energy consumption in all the simulated scenarios. The trickle timer send intervals which is responsible for sending DIO messages at intervals has been modified in this work, and the OF within the DIO message format itself which was encoded with RSSI readings in other to improve the efficiency and responsiveness of the protocol. This work developed different mathematical models for validation of simulation results and were discussed extensively in the work.

The routing approaches proposed in this work can be implemented in all the aspects of IoT applications discussed in chapter 3 of this work which can include healthcare, smart environment, smart agriculture, industrial and military applications. It is imperative to design an RPL routing protocol for Low powered devices that is reliably energy efficient in addition to its interoperability capabilities among devices with different standards. This work presented several proposed implementations of RPL routing protocol in literature, although most proposed protocols pose compatibility issues after modifying the DIO message. This compatibility issue is more rampant in mobile nodes because RPL has no support for mobility management which is the main disadvantage of RPL routing protocol.

Although energy consumption and QoS are metrics that present a satisfactory performance in RPL using its objective functions which is both flexible and compatible in a static network. Similarly, congestion is another issue of RPL that is still widely researched on despite several efforts to manage this issue some of which are discussed in chapter 3 of this work, there is still need to further exploit the standards and mechanisms to curtail this issue. Also, security is also another major concern in RPL routing that needs to be handled from the application layer down to the physical layer. Additionally, several efforts were made to handle mobility in RPL which were extensively discussed in chapters 3 and 4, where most of the proposed protocols showed a level of acceptable success and efficiency. However, after several tries on implementing the listed thresholds in chapter 6, this work was not able to achieve all, only the first threshold in each category was implemented in this thesis, the rest were left for future research.

It is strongly recommended that the adaptive fuzzy logic objective function be implemented due its generality in handling different routing situations, its high packet delivery ratio, delay tolerance, and enhanced energy efficiency.

7.2 Future Work

Although this research developed an adaptive routing protocol using fuzzy logic design, there are still so many issues that needs to be properly addressed in the future. A summary of the future work is presented as thus:

- Improving the memory and computational capacity of the COOJA simulator in Contiki OS in other to enable it handle more computational complexities.
- The presence of false positive selection of routing protocols which is mostly produced by the COOJA simulator module, whereby it gives false RSSI readings invoking the enhanced mobility routing when a node is static.
- Implementation of security at different layers to enable reliability and authentication in RPL routing protocol particularly in adaptive fuzzy logic objective function to ascertain the effect it will have on network overhead. Similarly, to test OR-OF with both TCP connection and security enabled RPL network configurations to determine the performance.
- The research intends to implement the objective functions with higher data rates of 8 packet/second, 16 packet/second, etc., as higher data rates cause more noise and congestion in the network.
- Deploying OR-OF on real life IoT applications which can allow different applications to communicate effectively and to analyse how the OF can accommodate high traffic load. The OR-OF allows traffic to reach the root node from all directions as opposed to the standardised versions of RPL that only allows traffic movement upwards. Also, allowing the OF to run for a long period to ascertains its effectiveness and availability as real practical data provides all the necessary information required to draw conclusions on its suitability.

References

Abreu, C., Ricardo, M., & Mendes, P. (2014). Energy-aware routing for biomedical wireless sensor networks. *Journal of Network and Computer Applications, 40*, 270-278.

Aburdene, M. F., & Goodman, T. J. (2005). Probability, Computer Networks, and Simulation. *age, 10*, 1.

Agarkhed, D. J., & Kadrolli, V. (2017). Fuzzy logic based Cluster Routing Techniques in WSN. *Proceedings in International Journal of Control Theory and Applications (IJCTA), 10*(9).

Ahmed, H. I., Nasr, A. A., Abdel-Mageid, S. M., & Aslan, H. K. (2021). DADEM: Distributed Attack Detection Model Based on Big Data Analytics for the Enhancement of the Security of Internet of Things (IoT). *International Journal of Ambient Computing and Intelligence (IJACI), 12*(1), 114-139.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials, 17*(4), 2347-2376.

Al-Kaseem, B. R., Al-Dunainawi, Y., & Al-Raweshidy, H. S. (2018). End-to-End Delay Enhancement in 6LoWPAN Testbed Using Programmable Network Concepts. *IEEE Internet of Things Journal, 6*(2), 3070-3086.

Al-Kashoash, H. A., Amer, H. M., Mihaylova, L., & Kemp, A. H. (2017). Optimization-based hybrid congestion alleviation for 6LoWPAN networks. *IEEE Internet of Things Journal, 4*(6), 2070-2081.

Al-Kashoash, H. A., Hafeez, M., & Kemp, A. H. (2017). Congestion control for 6LoWPAN networks: A game theoretic framework. *IEEE Internet of Things Journal, 4*(3), 760-771.

Al-Nidawi, Y., Yahya, H., & Kemp, A. H. (2015). *Impact of mobility on the IoT MAC infrastructure: IEEE 802.15. 4e TSCH and LLDN platform.* Paper presented at the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT).

Aliyu, U., Takruri, H., Hope, M., & Halilu, A. G. (2020). *DS-OLSR–Disaster Scenario Optimized Link State Routing Protocol.* Paper presented at the 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP).

Alliance, H. (2012). HomePlug GreenPHY v1. 1. *Reterived from* http://www. *homeplug. org/tech-resources/resources*.

ALMomani, I. M., & Saadeh, M. K. (2011). FEAR: Fuzzy-based energy aware routing protocol for wireless sensor networks. *International Journal of Communications, Network and System Sciences, 4*(06), 403.

Alsamhi, S. H., Afghah, F., Sahal, R., Hawbani, A., Al-qaness, A., Lee, B., & Guizani, M. (2021). Green internet of things using UAVs in B5G networks: A review of applications and strategies. *Ad Hoc Networks*, 102505.

Alves, R. C., Gabriel, L. B., de Oliveira, B. T., Margi, C. B., & dos Santos, F. C. L. (2015). Assisting physical (hydro) therapy with wireless sensors networks. *IEEE Internet of Things Journal, 2*(2), 113-120.

Anand, M. R., & Tahiliani, M. P. (2016). *mrpl++: Smarter-hop for optimizing mobility in rpl.* Paper presented at the 2016 IEEE Region 10 Symposium (TENSYMP).

Ancillotti, E., Bruno, R., Conti, M., Mingozzi, E., & Vallati, C. (2014). *Trickle-L 2: Lightweight link quality estimation through Trickle in RPL networks.* Paper presented at the Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014.

Ansere, J. A., Han, G., Liu, L., Peng, Y., & Kamal, M. (2020). Optimal Resource Allocation in Energy Efficient Internet of Things Networks with Imperfect CSI. *IEEE Internet of Things Journal*.

Baccelli, E., Hahm, O., Gunes, M., Wahlisch, M., & Schmidt, T. C. (2013). *RIOT OS: Towards an OS for the Internet of Things.* Paper presented at the 2013 IEEE conference on computer communications workshops (INFOCOM WKSHPS).

Bai, F., & Helmy, A. (2004). A survey of mobility models. *Wireless Adhoc Networks. University of Southern California, USA, 206*, 147.

Bai, F., & Helmy, A. (2008). Chapter 1 a survey of mobility models in wireless adhoc networks. *CiteSeerX-Scientific Literature Digital Library and Search Engine*.

Bandyopadhyay, S., & Coyle, E. J. (2003). *An energy efficient hierarchical clustering algorithm for wireless sensor networks.* Paper presented at the IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428).

Banh, M., Nguyen, N., Phung, K.-H., Nguyen, L., Thanh, N. H., & Steenhaut, K. (2016). *Energy balancing RPL-based routing for Internet of Things.* Paper presented at the 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE).

Barcelo, M., Correa, A., Vicario, J. L., Morell, A., & Vilajosana, X. (2015). Addressing mobility in RPL with position assisted metrics. *IEEE sensors journal, 16*(7), 2151-2161.

Barthel, D., Pister, K., Dejean, N., Vasseur, J., & Kim, M. (2012). Routing metrics used for path calculation in low-power and lossy networks.

Behera, T. M., Mohapatra, S. K., Samal, U. C., Khan, M. S., Daneshmand, M., & Gandomi, A. H. (2019). Residual energy-based cluster-head selection in WSNs for IoT application. *IEEE Internet of Things Journal, 6*(3), 5132-5139.

Bildea, A. (2013). *Link quality in wireless sensor networks.*

Boano, C. A., Tsiftes, N., Voigt, T., Brown, J., & Roedig, U. (2009). The impact of temperature on outdoor industrial sensornet applications. *IEEE Transactions on Industrial Informatics, 6*(3), 451-459.

Boris B., S. O. (2015). *Analysis of Packet Queuing in Telecommunication Networks*. Course Notes. (BSc.) Network Engineering, 2nd Year.

Bucchiarone, A., Battisti, S., Dias, T. G., & Feldman, P. (2021). Guest Editorial Diversification in Urban Transportation Systems and Beyond: Integrating People and Goods for the Future of Mobility. *IEEE Transactions on Intelligent Transportation Systems, 22*(4), 2008-2012.

Bush, S. (2015). Dect/ule connects homes for iot. *Electronics weekly, Retrieved from february, 6*, 2017.

Caicedo-Ortiz, J. G., De-la-Hoz-Franco, E., Ortega, R. M., Piñeres-Espitia, G., Combita-Niño, H., Estevez, F., & Cama-Pinto, A. (2018). Monitoring system for agronomic variables based in WSN technology on cassava crops. *Computers and Electronics in Agriculture, 145*, 275-281.

Casari, P., Castellani, A., Cenedese, A., Lora, C., Rossi, M., Schenato, L., & Zorzi, M. (2009). The "wireless sensor networks for city-wide ambient intelligence (WISE-WAI)" project. *Sensors, 9*(6), 4056-4082.

Castellani, A. P., Rossi, M., & Zorzi, M. (2014). Back pressure congestion control for CoAP/6LoWPAN networks. *Ad Hoc Networks, 18*, 71-84.

Celebican, O., Rosing, T. S., & Mooney III, V. J. (2004). *Energy estimation of peripheral devices in embedded systems.* Paper presented at the Proceedings of the 14th ACM Great Lakes symposium on VLSI.

Cetinkaya, O., & Akan, O. B. (2015). *A DASH7-based power metering system.* Paper presented at the 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC).

Chang, W.-W., Sung, T.-J., Huang, H.-W., Hsu, W.-C., Kuo, C.-W., Chang, J.-J., . . . Lin, Y.-Y. (2011). A smart medication system using wireless sensor network technologies. *Sensors and Actuators A: Physical, 172*(1), 315-321.

Chauvenet, C., Tourancheau, B., Genon-Catalot, D., Goudet, P.-E., & Pouillot, M. (2010). *A communication stack over PLC for multi physical layer IPv6 Networking.* Paper presented at the 2010 First IEEE International Conference on Smart Grid Communications.

Chekka, R. T., Miao, T., & Kim, K.-H. (2014). *Implementation of adaptive binary exponential backoff (ABEB) algorithm with dynamical sizing buffer for load-balanced RPL.* Paper presented at the 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN).

Chen, J., Kwong, K., Chang, D., Luk, J., & Bajcsy, R. (2006). *Wearable sensors for reliable fall detection.* Paper presented at the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference.

Chen, R., Guo, J., & Bao, F. (2014). Trust management for SOA-based IoT and its application to service composition. *IEEE Transactions on Services Computing, 9*(3), 482-495.

Chippalkatti, V. S. (2021). Review of Satellite based Internet of Things and Applications. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12*(12), 758-766.

Clausen, T., Yi, J., & Herberg, U. (2017). Lightweight on-demand ad hoc distance-vector routing-next generation (LOADng): Protocol, extension, and applicability. *Computer Networks, 126*, 125-140.

Cobarzan, C., Montavont, J., & Noel, T. (2014). *Analysis and performance evaluation of RPL under mobility.* Paper presented at the 2014 IEEE symposium on computers and communications (ISCC).

ContikiNG. (2019). Contiki New Generation. [Online].

Cruz, F. R. G., Sejera, M. P., Bunnao, M. B. G., Jovellanos, B. R., Maaño, P. L. C., & Santos, C. J. R. (2017). *Fall detection wearable device interconnected through zigbee network.* Paper presented at the 2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM).

Dawans, S., Duquennoy, S., & Bonaventure, O. (2012). *On link estimation in dense RPL deployments.* Paper presented at the 37th Annual IEEE Conference on Local Computer Networks-Workshops.

De Guglielmo, D., Brienza, S., & Anastasi, G. (2016). IEEE 802.15. 4e: A survey. *Computer Communications, 88*, 1-24.

De La Concepción, M. Á., Morillo, L. S., Gonzalez-Abril, L., & Ramírez, J. O. (2014). Discrete techniques applied to low-energy mobile human activity recognition. A new approach. *Expert Systems with Applications, 41*(14), 6138-6146.

de Waal, C., & Gerharz, M. (2003). Bonnmotion: A mobility scenario generation and analysis tool. In.

Di Marco, P., Athanasiou, G., Mekikis, P.-V., & Fischione, C. (2016). MAC-aware routing metrics for the internet of things. *Computer Communications, 74*, 77-86.

Dujovne, D., Watteyne, T., Vilajosana, X., & Thubert, P. (2014). 6TiSCH: deterministic IP-enabled industrial internet (of things). *IEEE Communications Magazine, 52*(12), 36-41.

El Korbi, I., Brahim, M. B., Adjih, C., & Saidane, L. A. (2012). *Mobility enhanced RPL for wireless sensor networks.* Paper presented at the 2012 Third international conference on the network of the future (NOF).

Fensli, R., Gunnarson, E., & Gundersen, T. (2005). *A wearable ECG-recording system for continuous arrhythmia monitoring in a wireless tele-home-care situation.* Paper presented at the 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05).

Gaddour, O., & Koubâa, A. (2012). RPL in a nutshell: A survey. *Computer Networks, 56*(14), 3163-3178.

Gaddour, O., Koubâa, A., & Abid, M. (2015). Quality-of-service aware routing for static and mobile IPv6-based low-power and lossy sensor networks using RPL. *Ad Hoc Networks, 33*, 233-256.

Gaddour, O., Koubâa, A., Baccour, N., & Abid, M. (2014). *OF-FL: QoS-aware fuzzy logic objective function for the RPL routing protocol.* Paper presented at the 2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt).

Gao, T., Pesto, C., Selavo, L., Chen, Y., Ko, J., Lim, J., . . . Chen, B.-r. (2008). *Wireless medical sensor networks in emergency response: Implementation and pilot results.* Paper presented at the 2008 IEEE Conference on Technologies for Homeland Security.

Gara, F., Saad, L. B., Ayed, R. B., & Tourancheau, B. (2015). *RPL protocol adapted for healthcare and medical applications.* Paper presented at the 2015 International wireless communications and mobile computing conference (IWCMC).

Gartner. (2014). *Gartner's 2014 hype cycle for emerging technologies maps the journey to digital business*. Retrieved from STAMFORD, USA:

Ghataoura, D. S., Yang, Y., & Matich, G. (2009). *GAFO: genetic adaptive fuzzy hop selection scheme for wireless sensor networks.* Paper presented at the Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly.

Ghazi Amor, M., Koubâa, A., Tovar, E., & Khalgui, M. (2016). *Cyber-OF: An adaptive cyber-physical objective function for smart cities applications.* Paper presented at the 28th Euromicro Conference on Real-Time Systems.

Gia, T. N., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. (2018). Fog computing approach for mobility support in internet-of-things systems. *IEEE Access, 6*, 36064-36082.

Gnawali, O. (2012). The minimum rank with hysteresis objective function.

Golpîra, H., Khan, S. A. R., & Safaeipour, S. (2021). A review of logistics internet-of-things: Current trends and scope for future research. *Journal of Industrial Information Integration*, 100194.

Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors, 12*(9), 11734-11753.

Gomez, C., Paradells, J., Bormann, C., & Crowcroft, J. (2017). From 6LoWPAN to 6Lo: Expanding the universe of IPv6-supported technologies for the Internet of Things. *IEEE Communications Magazine, 55*(12), 148-155.

Goyal, D., & Tripathy, M. R. (2012). *Routing protocols in wireless sensor networks: A survey.* Paper presented at the 2012 Second International Conference on Advanced Computing & Communication Technologies.

Gozuacik, N., & Oktug, S. (2015). Parent-aware routing for iot networks. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems* (pp. 23-33): Springer.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems, 29*(7), 1645-1660.

Gupta, I., Riordan, D., & Sampalli, S. (2005). *Cluster-head election using fuzzy logic for wireless sensor networks.* Paper presented at the CNSR.

Ha, M., Kwon, K., Kim, D., & Kong, P.-Y. (2014). *Dynamic and distributed load balancing scheme in multi-gateway based 6LoWPAN.* Paper presented at the 2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom).

Halilu, A. G., & Hope, M. (2021). *Energy Efficiency in Low power and Lossy Networks.* Paper presented at the 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA).

Halilu, A. G., Hope, M., Takruri, H., & Aliyu, U. (2020). *Optimized QoS Routing Protocol for Energy Scavenging Nodes in IoT.* Paper presented at the 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP).

Halilu, A. G., Hope, M., Takruri, H., & Aliyu, U. (2020). Optimized QoS Routing Protocol for Energy Scavenging Nodes in IoT. *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 1-6.

Hasan, M., Hossain, E., & Niyato, D. (2013). Random access for machine-to-machine communication in LTE-advanced networks: Issues and approaches. *IEEE Communications Magazine, 51*(6), 86-93.

Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). *Energy-efficient communication protocol for wireless microsensor networks.* Paper presented at the Proceedings of the 33rd annual Hawaii international conference on system sciences.

Heinzelman, W. R., Kulik, J., & Balakrishnan, H. (1999). *Adaptive protocols for information dissemination in wireless sensor networks.* Paper presented at the Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking.

Hellaoui, H., & Koudil, M. (2015). *Bird flocking congestion control for CoAP/RPL/6LoWPAN networks.* Paper presented at the Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems.

Hosseinzadeh, M., Koohpayehzadeh, J., Bali, A. O., Asghari, P., Souri, A., Mazaherinezhad, A., . . . Rawassizadeh, R. (2021). A diagnostic prediction model for chronic kidney disease in internet of things platform. *Multimedia Tools and Applications, 80*(11), 16933-16950.

Huang, J., Wu, X., Huang, W., Wu, X., & Wang, S. (2021). Internet of things in health management systems: A review. *International Journal of Communication Systems, 34*(4), e4683.

Hui, J. W. (2012). The routing protocol for low-power and lossy networks (rpl) option for carrying rpl information in data-plane datagrams.

Intanagonwiwat, C., Govindan, R., & Estrin, D. (2000). *Directed diffusion: A scalable and robust communication paradigm for sensor networks.* Paper presented at the Proceedings of the 6th annual international conference on Mobile computing and networking.

Iova, O., Picco, P., Istomin, T., & Kiraly, C. (2016). Rpl: The routing standard for the internet of things... or is it? *IEEE Communications Magazine, 54*(12), 16-22.

Iova, O., Theoleyre, F., & Noel, T. (2015a). *Exploiting multiple parents in RPL to improve both the network lifetime and its stability.* Paper presented at the 2015 IEEE International Conference on Communications (ICC).

Iova, O., Theoleyre, F., & Noel, T. (2015b). Using multiparent routing in RPL to increase the stability and the lifetime of the network. *Ad Hoc Networks, 29*, 45-62.

ITU, T. REC-G. 9959: Short Range Narrow-Band Digital Radiocommunication Transceivers—PHY, MAC, SAR and LLC Layer Specifications. In.

Iwanicki, K. (2016). *RNFD: Routing-layer detection of DODAG (root) node failures in low-power wireless networks.* Paper presented at the Proceedings of the 15th International Conference on Information Processing in Sensor Networks.

Javaid, M., & Khan, I. H. (2021). Internet of Things (IoT) enabled healthcare helps to take the challenges of COVID-19 Pandemic. *Journal of Oral Biology and Craniofacial Research, 11*(2), 209-214.

Javaid, N., Faisal, S., Khan, Z. A., Nayab, D., & Zahid, M. (2013). *Measuring fatigue of soldiers in wireless body area sensor networks.* Paper presented at the 2013 Eighth

International Conference on Broadband and Wireless Computing, Communication and Applications.

Jiang, J.-A., Tseng, C.-L., Lu, F.-M., Yang, E.-C., Wu, Z.-S., Chen, C.-P., . . . Liao, C.-S. (2008). A GSM-based remote wireless automatic monitoring system for field information: A case study for ecological monitoring of the oriental fruit fly, Bactrocera dorsalis (Hendel). *Computers and Electronics in Agriculture, 62*(2), 243-259.

Jiang, P., Xia, H., He, Z., & Wang, Z. (2009). Design of a water environment monitoring system based on wireless sensor networks. *Sensors, 9*(8), 6411-6434.

Kalmar, A., Vida, R., & Maliosz, M. (2015). *Caesar: A context-aware addressing and routing scheme for rpl networks.* Paper presented at the 2015 IEEE International Conference on Communications (ICC).

Kamenar, E., Zelenika, S., Blažević, D., Maćešić, S., Gregov, G., Marković, K., & Glažar, V. (2016). Harvesting of river flow energy for wireless sensor network technology. *Microsystem Technologies, 22*(7), 1557-1574.

Kamgueu, P.-O., Nataf, E., & Djotio, T. N. (2015). *On design and deployment of fuzzy-based metric for routing in low-power and lossy networks.* Paper presented at the 2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops).

Kamgueu, P. O., Nataf, E., Ndié, T. D., & Festor, O. (2013). Energy-based routing metric for RPL.

Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F., & Alonso-Zarate, J. (2015). A survey on application layer protocols for the internet of things. *Transaction on IoT and Cloud computing, 3*(1), 11-17.

Karkazis, P., Leligou, H. C., Sarakis, L., Zahariadis, T., Trakadas, P., Velivassaki, T. H., & Capsalis, C. (2012). *Design of primary and composite routing metrics for RPL-compliant wireless sensor networks.* Paper presented at the 2012 International Conference on Telecommunications and Multimedia (TEMU).

Khalil, N., Abid, M. R., Benhaddou, D., & Gerndt, M. (2014). *Wireless sensors networks for Internet of Things.* Paper presented at the 2014 IEEE ninth international conference on Intelligent sensors, sensor networks and information processing (ISSNIP).

Khan, M. M., Lodhi, M. A., Rehman, A., Khan, A., & Hussain, F. B. (2016). Sink-to-sink coordination framework using RPL: Routing protocol for low power and lossy networks. *Journal of Sensors, 2016*.

Kharrufa, H., Al-Kashoash, H., Al-Nidawi, Y., Mosquera, M. Q., & Kemp, A. H. (2017). *Dynamic RPL for multi-hop routing in IoT applications.* Paper presented at the 2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS).

Kharrufa, H., Al-Kashoash, H., & Kemp, A. H. (2018). A game theoretic optimization of RPL for mobile Internet of Things applications. *IEEE sensors journal, 18*(6), 2520-2530.

Kharrufa, H. D. Y. (2018). *Data Routing for Mobile Internet of Things Applications.* University of Leeds,

Khelifi, N., Oteafy, S., Hassanein, H., & Youssef, H. (2015). *Proactive maintenance in RPL for 6LowPAN.* Paper presented at the 2015 International Wireless Communications and Mobile Computing Conference (IWCMC).

Kim, A. N., Hekland, F., Petersen, S., & Doyle, P. (2008). *When HART goes wireless: Understanding and implementing the WirelessHART standard.* Paper presented at the 2008 IEEE International Conference on Emerging Technologies and Factory Automation.

Kim, H.-S., Kim, H., Paek, J., & Bahk, S. (2016). Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks. *IEEE Transactions on mobile computing, 16*(4), 964-979.

Ko, J., & Chang, M. (2014). Momoro: Providing mobility support for low-power wireless applications. *IEEE Systems Journal, 9*(2), 585-594.

Ko, J., Lu, C., Srivastava, M. B., Stankovic, J. A., Terzis, A., & Welsh, M. (2010). Wireless sensor networks for healthcare. *Proceedings of the IEEE, 98*(11), 1947-1960.

Kryszkiewicz, P. (2020). *Neuron-Inspired Communications for Energy Efficient Internet of Things Networks.* Paper presented at the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops).

Kushalnagar, N., & Montenegro, G. (2007). Transmission of IPv6 packets over IEEE 802.15. 4 networks.

Kwong, K. H., Wu, T.-T., Goh, H. G., Sasloglou, K., Stephen, B., Glover, I., . . . Andonovic, I. (2012). Practical considerations for wireless sensor networks in cattle monitoring applications. *Computers and Electronics in Agriculture, 81*, 33-44.

Lakshman, S. A., & Ebenezer, D. (2021). Integration of internet of things and drones and its future applications. *Materials Today: Proceedings*.

Lamaazi, H., & Benamar, N. (2018). OF-EC: A novel energy consumption aware objective function for RPL based on fuzzy logic. *Journal of Network and Computer Applications, 117*, 42-58.

Lamaazi, H., Benamar, N., Imaduddin, M. I., & Jara, A. J. (2015). *Performance assessment of the routing protocol for low power and lossy networks.* Paper presented at the 2015 International Conference on Wireless Networks and Mobile Communications (WINCOM).

Le, Q., Ngo-Quynh, T., & Magedanz, T. (2014). *Rpl-based multipath routing protocols for internet of things on wireless sensor networks.* Paper presented at the 2014 International Conference on Advanced Technologies for Communications (ATC 2014).

Lee, J., & Chung, K. (2011). An efficient transmission power control scheme for temperature variation in wireless sensor networks. *Sensors, 11*(3), 3078-3093.

Lekidis, A., & Katsaros, P. (2018). Model-based design of energy-efficient applications for IoT systems. *arXiv preprint arXiv:1807.01242*.

Levis, P., & Clausen, T. H. (2011). The trickle algorithm.

Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., . . . Brewer, E. (2005). TinyOS: An operating system for sensor networks. In *Ambient intelligence* (pp. 115-148): Springer.

Li, G., Ma, M., Liu, C., & Shu, Y. (2017). Adaptive fuzzy multiple attribute decision routing in VANETs. *International Journal of Communication Systems, 30*(4), e3014.

Li, J., Dai, J., Issakhov, A., Almojil, S. F., & Souri, A. (2021). Towards decision support systems for energy management in the smart industry and Internet of Things. *Computers & Industrial Engineering, 161*, 107671.

Li, Y., Li, D., Cui, W., & Zhang, R. (2011). *Research based on OSI model.* Paper presented at the 2011 IEEE 3rd International Conference on Communication Software and Networks.

Lindsey, S., & Raghavendra, C. S. (2002). *PEGASIS: Power-efficient gathering in sensor information systems.* Paper presented at the Proceedings, IEEE aerospace conference.

Liu, L., Zhang, T., & Zhang, J. (2013). DAG based multipath routing algorithm for load balancing in machine-to-machine networks. *International Journal of Distributed Sensor Networks, 10*(1), 457962.

Malik, P. K., Sharma, R., Singh, R., Gehlot, A., Satapathy, S. C., Alnumay, W. S., . . . Nayak, J. (2021). Industrial Internet of Things and its applications in industry 4.0: State of the art. *Computer Communications, 166*, 125-139.

Mascolo, C., & Musolesi, M. (2006). *SCAR: context-aware adaptive routing in delay tolerant mobile sensor networks.* Paper presented at the Proceedings of the 2006 international conference on Wireless communications and mobile computing.

Mayzaud, A., Badonnel, R., & Chrisment, I. (2017). A distributed monitoring strategy for detecting version number attacks in RPL-based networks. *IEEE Transactions on Network and Service Management, 14*(2), 472-486.

Meddeb, M., Dhraief, A., Belghith, A., Monteil, T., Drira, K., & Gannouni, S. (2018). AFIRM: Adaptive forwarding based link recovery for mobility support in NDN/IoT networks. *Future Generation Computer Systems, 87*, 351-363.

Michopoulos, V., Guan, L., Oikonomou, G., & Phillips, I. (2012). *DCCC6: Duty Cycle-aware congestion control for 6LoWPAN networks.* Paper presented at the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops.

Mohamed, B., & Mohamed, F. (2015). QoS routing RPL for low power and lossy networks. *International Journal of Distributed Sensor Networks, 11*(11), 971545.

Monshi, M. M., & Mohammed, O. A. (2013). *A study on the efficient wireless sensor networks for operation monitoring and control in smart grid applications.* Paper presented at the 2013 Proceedings of IEEE Southeastcon.

Mottola, L., Picco, G. P., Ceriotti, M., Gună, Ş., & Murphy, A. L. (2010). Not all wireless sensor networks are created equal: A comparative study on tunnels. *ACM Transactions on Sensor Networks (TOSN), 7*(2), 15.

Munusamy, A., Adhikari, M., Khan, M. A., Menon, V. G., Srirama, S. N., Alex, L. T., & Khosravi, M. R. (2021). Edge-centric secure service provisioning in IoT-Enabled maritime transportation systems. *IEEE Transactions on Intelligent Transportation Systems*.

Musolesi, M., Hailes, S., & Mascolo, C. (2004). *An ad hoc mobility model founded on social network theory.* Paper presented at the Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems.

Musolesi, M., & Mascolo, C. (2008). Car: context-aware adaptive routing for delay-tolerant mobile networks. *IEEE Transactions on mobile computing, 8*(2), 246-260.

Nassiri, M., Boujari, M., & Azhari, S. V. (2015). Energy-aware and load-balanced parent selection in RPL routing for wireless sensor networks. *International Journal of Wireless and Mobile Computing, 9*(3), 231-239.

Nesa, N., & Banerjee, I. (2018). SensorRank: An energy efficient sensor activation algorithm for sensor data fusion in wireless networks. *IEEE Internet of Things Journal, 6*(2), 2532-2539.

Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., & Gomez, C. (2015). Ipv6 over bluetooth (r) low energy. *RFC 7668*.

Niu, X., Huang, X., Zhao, Z., Zhang, Y., Huang, C., & Cui, L. (2007). *The design and evaluation of a wireless sensor network for mine safety monitoring.* Paper presented at the IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference.

Nurmio, J., Nigussie, E., & Poellabauer, C. (2015). *Equalizing energy distribution in sensor nodes through optimization of RPL.* Paper presented at the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing.

Park, M. (2015). IEEE 802.11 ah: sub-1-GHz license-exempt operation for the internet of things. *IEEE Communications Magazine, 53*(9), 145-151.

Patel, S., Lorincz, K., Hughes, R., Huggins, N., Growdon, J. H., Welsh, M., & Bonato, P. (2007). *Analysis of feature space for monitoring persons with Parkinson's disease with application to a wireless wearable sensor system.* Paper presented at the 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.

Pavkovic, B. (2012). *Going towards the future Internet of Things through a cross-layer optimization of the standard protocol suite.*

Pavković, B., Theoleyre, F., & Duda, A. (2011). *Multipath opportunistic RPL routing over IEEE 802.15. 4.* Paper presented at the Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems.

Pivoto, D. G., de Almeida, L. F., da Rosa Righi, R., Rodrigues, J. J., Lugli, A. B., & Alberti, A. M. (2021). Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of Manufacturing Systems, 58*, 176-192.

Poole, I. (2014). Weightless wireless—m2m white space communications-tutorial. In.

Popa, D., Salazar, R., Dejean, N., & Jetcheva, J. (2011). Applicability statement for the routing protocol for low power and lossy networks (rpl) in ami networks.

Poudel, S. (2016). Internet of things: underlying technologies, interoperability, and threats to privacy and security. *Berkeley Tech. LJ, 31*, 997.

Qaim, W. B., Ometov, A., Molinaro, A., Lener, I., Campolo, C., Lohan, E. S., & Nurmi, J. (2020). Towards Energy Efficiency in the Internet of Wearable Things: A Systematic Review. *IEEE Access*.

Qasem, M. (2018). *Load balancing and context aware enhancements for RPL routed Internet of Things.* Edinburgh Napier University,

Qasem, M., Al-Dubai, A., Romdhani, I., Ghaleb, B., & Gharibi, W. (2016). *A new efficient objective function for routing in Internet of Things paradigm.* Paper presented at the 2016 IEEE Conference on Standards for Communications and Networking (CSCN).

Qin, H., Li, Z., Wang, Y., Lu, X., Zhang, W., & Wang, G. (2010). *An integrated network of roadside sensors and vehicles for driving safety: Concept, design and experiments.* Paper presented at the 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom).

Raza, S., & Voigt, T. (2010). *Interconnecting WirelessHART and legacy HART networks.* Paper presented at the 2010 6th IEEE International Conference on Distributed Computing in Sensor Systems Workshops (DCOSSW).

Raza, S., Wallgren, L., & Voigt, T. (2013). SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Networks, 11*(8), 2661-2674.

Reimers, S. O. (2012). Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded Systems. *Synchrone Sprachen*, 5.

Ren, H., Meng, M. Q.-H., & Chen, X. (2006). *Wireless assistive sensor networks for the deaf.* Paper presented at the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Riker, A., Curado, M., & Monteiro, E. (2017). *Neutral operation of the minimum energy node in energy-harvesting environments.* Paper presented at the 2017 IEEE Symposium on Computers and Communications (ISCC).

Rodríguez-Molina, J., Bilbao, S., Martínez, B., Frasheri, M., & Cürüklü, B. (2017). An optimized, data distribution service-based solution for reliable data exchange among autonomous underwater vehicles. *Sensors, 17*(8), 1802.

Ropitault, T., Pelov, A., Toutain, L., Vedantham, R., & Itron, P. C. (2015). *Channel occupancy-Faster, better, stronger.* Paper presented at the 2015 IEEE International Conference on Smart Grid Communications (SmartGridComm).

Saad, L. B., Chauvenet, C., & Tourancheau, B. (2011). *Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: two cases studies*.

Saad, L. B., & Tourancheau, B. (2011). *Sinks mobility strategy in IPv6-based WSNs for network lifetime improvement.* Paper presented at the 2011 4th IFIP International Conference on New Technologies, Mobility and Security.

Safdar, V., Bashir, F., Hamid, Z., Afzal, H., & Pyun, J. Y. (2012). *A hybrid routing protocol for wireless sensor networks with mobile sinks.* Paper presented at the ISWPC 2012 proceedings.

Saint-Andre, P. (2011). Extensible messaging and presence protocol (XMPP): Core.

Salman, T., & Jain, R. (2015). Networking protocols and standards for internet of things. *Internet of Things and Data Analytics Handbook (2015), 7*.

Shakya, N. M., Mani, M., & Crespi, N. (2017). *SEEOF: Smart energy efficient objective function: Adapting RPL objective function to enable an IPv6 meshed topology solution for battery operated smart meters.* Paper presented at the 2017 Global Internet of Things Summit (GIoTS).

Sharkawy, B., Khattab, A., & Elsayed, K. M. (2014). *Fault-tolerant RPL through context awareness.* Paper presented at the 2014 IEEE World Forum on Internet of Things (WF-IoT).

Shelby, Z., Hartke, K., & Bormann, C. (2014). The constrained application protocol (CoAP).

Sheu, J.-P., Hsu, C.-X., & Ma, C. (2015). *A game theory based congestion control protocol for wireless personal area networks.* Paper presented at the 2015 IEEE 39th Annual Computer Software and Applications Conference.

Shin, J., & Suh, C. (2011). CREEC: Chain routing with even energy consumption. *Journal of Communications and Networks, 13*(1), 17-25.

Sikora, A., Schappacher, M., Zimmermann, L., & Mars, N. (2017). Development of Thread-compatible Open Source Stack.

Singh, M., Rajan, M., Shivraj, V., & Balamuralidhar, P. (2015). *Secure mqtt for internet of things (iot).* Paper presented at the 2015 Fifth International Conference on Communication Systems and Network Technologies.

Singh, R. K., Puluckul, P. P., Berkvens, R., & Weyn, M. (2020). Energy consumption analysis of LPWAN technologies and lifetime estimation for IoT application. *Sensors, 20*(17), 4794.

Sinha, S. R., & Park, Y. (2017). Building an IoT Ecosystem Framework. In *Building an Effective IoT Ecosystem for Your Business* (pp. 1-20): Springer.

Specification, W. (2008). 75: Tdma data-link layer. *HART Communication Foundation Std., Rev, 1*(1).

Srinivasan, K., Kazandjieva, M. A., Agarwal, S., & Levis, P. (2008). *The β-factor: measuring wireless link burstiness.* Paper presented at the Proceedings of the 6th ACM conference on Embedded network sensor systems.

Srinivasan, K., & Levis, P. (2006). *RSSI is under appreciated.* Paper presented at the Proceedings of the third workshop on embedded networked sensors (EmNets).

Standard, O. (2012). Oasis advanced message queuing protocol (amqp) version 1.0. *International Journal of Aerospace Engineering Hindawi www. hindawi. com, 2018*.

Stankovic, J. A. (2014). Research directions for the internet of things. *IEEE Internet of Things Journal, 1*(1), 3-9.

Taghizadeh, S., Bobarshad, H., & Elbiaze, H. (2018). CLRPL: context-aware and load balancing RPL for IoT networks under heavy and highly dynamic load. *IEEE Access, 6*, 23277-23291.

Tall, H., Chalhoub, G., & Misson, M. (2017). *M-CoLBA: Multichannel Collaborative Load Balancing Algorithm with queue overflow avoidance in WSNs.* Paper presented at the 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC).

Tang, L., Wang, K.-C., Huang, Y., & Gu, F. (2007). Channel characterization and link quality assessment of IEEE 802.15. 4-compliant radio for factory environments. *IEEE Transactions on Industrial Informatics, 3*(2), 99-110.

Tariq, A., Rehman, R. A., & Kim, B.-S. (2020). *Energy Efficient Priority Aware Forwarding in SDN Enabled Named Data Internet of Things.* Paper presented at the 2020 International Conference on Electronics, Information, and Communication (ICEIC).

Thubert, P. (2012). Objective function zero for the routing protocol for low-power and lossy networks (RPL).

Thubert, P., & Hui, J. W. (2011). Compression format for IPv6 datagrams over IEEE 802.15. 4-based networks.

Thubert, P., Pelov, A., & Krishnan, S. (2017). Low-power wide-area networks at the ietf. *IEEE Communications Standards Magazine, 1*(1), 76-79.

Tian, B., Hou, K. M., Shi, H., Liu, X., Diao, X., Li, J., . . . Chanet, J.-P. (2013). *Application of modified RPL under VANET-WSN communication architecture.* Paper presented at the 2013 international conference on computational and information sciences.

TinyOS. (2019). TnyOS Documentation Wiki [Online]. Retrieved from http://tinyos.stanford.edu/tinyos-wiki/index.php/Main_page

Tsiftes, N., Eriksson, J., & Dunkels, A. (2010). *Low-power wireless IPv6 routing with ContikiRPL.* Paper presented at the The 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).

Tsitsiashvili, G., & Osipova, M. (2018). Generalization and extension of Burke theorem. *Reliability: Theory & Applications, 13*(1 (48)).

Tubaishat, M., Zhuang, P., Qi, Q., & Shang, Y. (2009). Wireless sensor networks in intelligent transportation systems. *Wireless communications and mobile computing, 9*(3), 287-302.

Tun, S. Y. Y., Madanian, S., & Mirza, F. (2021). Internet of things (IoT) applications for elderly care: a reflective review. *Aging clinical and experimental research, 33*(4), 855-867.

Velmani, R., & Kaarthick, B. (2014). An efficient cluster-tree based data collection scheme for large mobile wireless sensor networks. *IEEE sensors journal, 15*(4), 2377-2390.

Villaverde, B. C., Alberola, R. D. P., Jara, A. J., Fedor, S., Das, S. K., & Pesch, D. (2013). Service discovery protocols for constrained machine-to-machine communications. *IEEE communications surveys & tutorials, 16*(1), 41-60.

Vučinić, M., Tourancheau, B., & Duda, A. (2013). *Performance comparison of the rpl and loadng routing protocols in a home automation scenario.* Paper presented at the 2013 IEEE Wireless Communications and Networking Conference (WCNC).

Winchcomb, T., Massey, S., & Beastall, P. (2017). Review of the latest development in the Internet of Things. *Cambridge Consultants, Ofcom contract number 1636 (MC370).*

Winter, T. (2012). RPL: IPv6 routing protocol for low-power and lossy networks.

Xia, F., Li, J., Hao, R., Kong, X., & Gao, R. (2013). Service differentiated and adaptive CSMA/CA over IEEE 802.15. 4 for cyber-physical systems. *The Scientific World Journal, 2013*.

Xu, Y., Yue, Z., & Lv, L. (2019). Clustering Routing Algorithm and Simulation of Internet of Things Perception Layer Based on Energy Balance. *IEEE Access, 7*, 145667-145676.

Yaïci, W., Krishnamurthy, K., Entchev, E., & Longo, M. (2020). *Internet of Things for Power and Energy Systems Applications in Buildings: An Overview.* Paper presented at the 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe).

Yang, L., Yang, S.-H., & Plotnick, L. (2013). How the internet of things technology enhances emergency response operations. *Technological Forecasting and Social Change, 80*(9), 1854-1867.

Yao, Y., & Gehrke, J. (2002). The cougar approach to in-network query processing in sensor networks. *ACM Sigmod record, 31*(3), 9-18.

Younis, O., & Fahmy, S. (2004). HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on mobile computing*(4), 366-379.

Yunis, J. P., & Dujovne, D. (2014). *Energy efficient routing performance evaluation for LLNs using combined metrics.* Paper presented at the 2014 IEEE Biennial Congress of Argentina (ARGENCON).

Zacharias, S., Newe, T., O'Keeffe, S., & Lewis, E. (2012). *Identifying sources of interference in RSSI traces of a single IEEE 802.15. 4 channel.* Paper presented at the The 8th International Conference on Wireless and Mobile Communications.

Zeynali, M., Khanli, L. M., & Mollanejad, A. (2009). *Edarp: novel energy and distance-aware routing protocol in wireless sensor network.* Paper presented at the Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human.

Zhen, L., Bashir, A. K., Yu, K., Al-Otaibi, Y. D., Foh, C. H., & Xiao, P. (2020). Energy-Efficient Random Access for LEO Satellite-Assisted 6G Internet of Remote Things. *IEEE Internet of Things Journal*.

Zhou, G., He, T., Krishnamurthy, S., & Stankovic, J. A. (2004). *Impact of radio irregularity on wireless sensor networks.* Paper presented at the Proceedings of the 2nd international conference on Mobile systems, applications, and services.

Zhu, L., Wang, R., & Yang, H. (2017). Multi-path data distribution mechanism based on rpl for energy consumption and time delay. *Information, 8*(4), 124.

Zuo, J. (2013). *Cross-layer aided routing design for ad hoc networks.* University of Southampton,

Appendices

Appendix 1: Psi values derived for Mathematical Modelling

ψ values						
	Mobility Models	***Energy_OF***	***Mobility_OF***	***QoS_OF***	***OF_0***	***MRHOF***
10 nodes	*No Mobility*	*0.425*	*0.475*	*0.3*	*0.530*	*0.520*
	RWK	*0.56*	*0.51*	*-*	*0.5745*	*0.533*
	RWP	*0.598*	*0.486*	*-*	*0.617*	*0.545*
20 nodes	*No Mobility*	*0.525*	*0.590*	*0.35*	*0.585*	*0.548*
	RWK	*0.633*	*0.568*	*-*	*0.6152*	*0.558*
	RWP	*0.627*	*0.551*	*-*	*0.639*	*0.599*
30 nodes	*No Mobility*	*0.575*	*0.625*	*0.4*	*0.632*	*0.580*
	RWK	*0.6912*	*0.611*	*-*	*0.652*	*0.593*
	RWP	*0.6973*	*0.6009*	*-*	*0.671*	*0.637*

Appendix 2: Mobility Function Computation (MATLAB Codes)

```
sum =0
a = 1;% Minimum speed at nodes
b = 4;% Maximum speed at nodes
nodes = 20 % Number of nodes considered (10, 20, 30)
V = (b-a).*rand(1,nodes)+a;% A vector representing speed at nodes btw 1 and
4
%n = length(V)
k = 2
no_pairednodes = nchoosek(nodes,k)% This calculates the total number of
paired nodes
for i = 1:(length(V)-1)
    for j = i+1: length(V)
        sum = sum + abs(V(i)-V(j));
    end
end
(1/no_pairednodes)*sum
```

Appendix 3: Method of Computation for Mathematical Models

Computed Results for Packet Delivery Ratio (%) 10 nodes							
	x_0	x_1	x_2	x_3	x_4	x_5	x_6
	ψ	${x_0}^3$	$(1 - x_1) * R_d$	S_t	x_2/x_3	$x_4 * 100$	PDR
Energy_OF	*0.425*	*0.07677*	*99,709.30*	*108,000*	*0.92323*	*92.323*	*92.32*
Mobility_OF	*0.475*	*0.10717*	*96,425.44*	*108,000*	*0.89283*	*89.283*	*89.28*
QoS_OF	*0.3*	*0.027*	*105,084*	*108,000*	*0.973*	*97.3*	*97.30*

Computed Results for End-to-End Delay (ms) 10 nodes								
	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
	ψ	$\lambda * y_0$	μ	$y_2 - y_1$	y_1/y_2	$1 + y_4$	y_5/y_3	D_{e2e}
Energy_OF	*0.425*	*27.69703*	*800*	*772.30297*	*0.034623*	*1.034623*	*0.001340*	*48.22766*
Mobility_OF	*0.475*	*26.78484*	*800*	*773.21516*	*0.033481*	*1.033481*	*0.001337*	*48.11768*
QoS_OF	*0.3*	*29.19*	*800*	*770.81*	*0.00134*	*1.00134*	*0.001345*	*24.20412*

Computed Results for Energy Consumption (mj) 10 nodes							
	z_0	z_1	z_2	z_3	z_4	z_5	z_6
	ψ	$3 * I_i$	t	$z_1 * z_2 * n$	z_3/B_{tt}	$z_4 * z_0 * E_t^i$	z_5 to mj
Energy_OF	*0.425*	*0.129135*	*1800*	*2324.43*	*0.070936*	*0.0226108*	*22.6108*
Mobility_OF	*0.475*	*0.129135*	*1800*	*2324.43*	*0.070936*	*0.0252709*	*25.2709*
QoS_OF	*0.3*	*0.129135*	*1800*	*2324.43*	*0.070936*	*0.0159605*	*15.9605*

Appendix 4: Network Routing Overhead Parameters and Equations

S/N	*Parameter*	*Function*
1.	$\boldsymbol{R_0}$	*Routing Overhead*
2.	$\boldsymbol{P_r}$	*Periodic message*
3.	$\boldsymbol{P_f}$	*Packet fails*
4.	$\boldsymbol{T_r}$	*Triggered message update*
5.	$\boldsymbol{K}$	*Routing overhead impulse factor (constant as 1)*
6.	$\boldsymbol{B}$	*Transmission Rate*
7.	$\boldsymbol{Q_r^l}$	*Probability that an uplink remains unchanged during r hops*
8.	$\boldsymbol{Na}$	*Number of arrived packets*
9.	$\boldsymbol{n}$	*Number of nodes*
10.	$\boldsymbol{T_{PR}}$	*Periodic route update interval*
11.	$\boldsymbol{\mu_k}$	*Uplink time*

$$R_o = P_r + P_f + T_r$$

$$P_r = \frac{kn^3}{BT_{pr}}$$

$$P_f = \left[\sum_{P_i^{\varepsilon PA}} \sum_{r=0}^{l_i} Q_r^l\ (T_{PR})\ Na\ (T_{PR})\right]$$

$$T_r = \sum_{i=1}^{n} \frac{\left\lceil \frac{T}{T_{pr}} \right\rceil}{\frac{T}{T_{pr}}}$$

$$Q_r^l\ (T_{PR}) = 1 - e^{\frac{rT_{pr}}{\mu_k}}$$

$$R_o = \frac{kn^3}{BT_{PR}} + \left[\sum_{P_i^{\varepsilon PA}} \sum_{r=0}^{l_i} 1 - e^{\frac{rT_{pr}}{\mu_k}}\ Na\ (T_{PR})\right] + \sum_{i=1}^{n} \frac{\left\lceil \frac{T}{T_{pr}} \right\rceil}{\frac{T}{T_{pr}}}\ i$$

Appendix 5: Modelled Results for Network Performance

Modelled Results of Packet Delivery Ratio

Packet Delivery Ratio (%)									
	10 nodes			**20 nodes**			**30 nodes**		
	No mobility	**RWK**	**RWP**	**No Mobility**	**RWK**	**RWP**	**No Mobility**	**RWK**	**RWP**
rpl_energyof	92.8042	78.4058	82.5041	86.9998	74.6487	75.3846	80.7800	66.9927	69.5847
(Energy_computed)	**92.32**	**-**	**-**	**85.530**	**-**	**-**	**80.99**	**-**	**-**
rpl_mobilityof	74.5278	86.95121	88.5789	71.64112	81.76345	83.33223	68.27833	77.109	78.4523
(Mobility_computed)	**89.28**	**-**	**-**	**79.46**	**-**	**-**	**75.59**	**-**	**-**
rpl_QoSof	98.23	-	-	95.22	-	-	91.67	-	-
(QoS_computed)	**97.30**	**-**	**-**	**95.7125**	**-**	**-**	**93.6**	**-**	**-**

Modelled Results of End-to-End Delay

End-to-End Delay (ms)									
	10 nodes			**20 nodes**			**30 nodes**		
	No mobility	**RWK**	**RWP**	**No Mobility**	**RWK**	**RWP**	**No Mobility**	**RWK**	**RWP**
rpl_energyof	37.38901	46.44779	45.78	41.8118	59.53551	58.24	47.10804	77.64655	79.02
(Energy_computed)	**48.22765**	**-**	**-**	**57.56509**	**-**	**-**	**67.5277**	**-**	**-**
rpl_mobilityof	-	47.31	45.79	-	53.57	50.39	-	59.32	55.61
(Mobility_computed)	**48.11768**	**-**	**-**	**57.04155**	**-**	**-**	**66.7054**	**-**	**-**
rpl_staticof	17.29	-	-	24.25	-	-	29.37	-	-
(QoS_computed)	**24.2364**	**-**	**-**	**31.17614**	**-**	**-**	**42.1618**	**-**	**-**

Modelled Results of Energy Consumption

Energy Consumption (mj)									
	10 nodes			**20 nodes**			**30 nodes**		
	No mobility	**RWK**	**RWP**	**No Mobility**	**RWK**	**RWP**	**No Mobility**	**RWK**	**RWP**
rpl_energyof	22.1	33.2602	32.9409	30.7	41.6308	38.8531	38.4	53.7292	54.2117
(Energy_computed)	**22.6108**	**29.7931**	**31.8148**	**37.4416**	**45.4638**	**45.0328**	**46.1337**	**55.8493**	**56.3422**
rpl_mobilityof	-	32.4	28.9	-	45.9	37.6	-	51.4	44.8
(Mobility_computed)	**-**	**27.1330**	**25.8562**	**-**	**40.79528**	**39.2961**	**-**	**49.3691**	**48.2117**
rpl_staticof	13.3	-	-	18.2	-	-	22.343	-	-
(QoS_computed)	**15.9606**	**-**	**-**	**24.9612**	**-**	**-**	**32.093**	**-**	**-**

Appendix 6: Simulation Results for Network Performance

Simulation Results for PDR (%)

	Mobility Models	*Energy_OF*	*Mobility_OF*	*QoS_OF*	*OF_0*	*MRHOF*
10 nodes	*No Mobility*	*92.8042*	*74.5278*	98.23	*79.54766*	*85.92349*
	Random Walk	*78.40576*	*86.95121*	-	*63.02598*	*84.94104*
	Random W/P	*82.50408*	*88.5789*	-	*63.18579*	*83.85324*
20 nodes	*No Mobility*	*86.99977*	*71.64112*	*95.22*	*73.97996*	*83.59688*
	Random Walk	*74.64865*	*81.76345*	-	*56.78124*	*82.60534*
	Random W/P	*75.38457*	*83.33223*	-	*58.39729*	*78.55258*
30 nodes	*No Mobility*	*80.78001*	*68.27833*	*91.67*	*67.85313*	*80.42612*
	Random Walk	*66.99268*	*77.109*	-	*50.31306*	*79.18625*
	Random W/P	*69.58468*	*78.4523*	-	*51.5329*	*74.16296*

Simulation Results for End − to − End Delay (ms)

	Mobility Models	*Energy_OF*	*Mobility_OF*	*QoS_OF*	*OF_0*	*MRHOF*
10 nodes	*No Mobility*	*37.38901*	-	*17.29*	*46.96283*	*41.34136*
	Random Walk	*46.44779*	*47.31*	-	*54.6553*	*60.96283*
	Random W/P	*45.78*	*45.79*	-	*52.54604*	*56.54604*
20 nodes	*No Mobility*	*41.8118*	-	*24.25*	*5.3.76854*	*47.40331*
	Random Walk	*59.53551*	*53.57*	-	*63.27214*	*70.76854*
	Random W/P	*58.24*	*50.39*	-	*59.17069*	*69.17069*
30 nodes	*No Mobility*	*47.10804*	-	*29.37*	*68.24794*	*57.26136*
	Random Walk	*77.64655*	*59.32*	-	*75.83797*	*81.24794*
	Random W/P	*79.02*	*55.61*	-	*70.12555*	*77.12555*

Simulation Results for Energy Consumption (mj)

	Mobility Models	*Energy_OF*	*Mobility_OF*	*QoS_OF*	*OF_0*	*MRHOF*
10 nodes	*No Mobility*	*22.3*	-	*13.3*	*31.84847*	*30.46936*
	Random Walk	*33.2602*	*32.4*	-	*37.09287*	*32.29944*
	Random W/P	*28.94095*	*28.9*	-	*34.19364*	*38.46662*
20 nodes	*No Mobility*	*24.6*	-	*18.2*	*33.60119*	*31.14756*
	Random Walk	*35.63081*	*45.9*	-	*40.4163*	*38.14235*
	Random W/P	*30.85311*	*32.6*	-	*54.51584*	*45.63479*
30 nodes	*No Mobility*	*28.4*	-	*22.3*	*41.43932*	*38.61883*
	Random Walk	*37.72917*	*51.4*	-	*51.4536*	*49.3376*
	Random W/P	*33.21169*	*39.7*	-	*66.96205*	*63.72185*

Appendix 7: Method of Metric Combination to a Single Readable Value

Below is an example of metric combination using linear combination after the fuzzy logic has been applied to turn the result into a single readable value for decision making purposes. The processes of using fuzzification, fuzzy inference engine and defuzzification will be applied in the following example using metric combination of *rpl_mobilityof* (hop count and delay metrics).

Consider a route where the number of hops is 3 (average number of hops in research simulation), hence the normalized number of hops is $hp_{nor} = \frac{3}{10} = 0.3$, where the maximum number of hops set for research simulation is $H = 10$. The latency set for the route-lifetime is given by $l_t = 20ms$, then membership functions as described in chapter 3.9 from equations 3.1 – 3.15 which can be vividly described as in Figure 1,

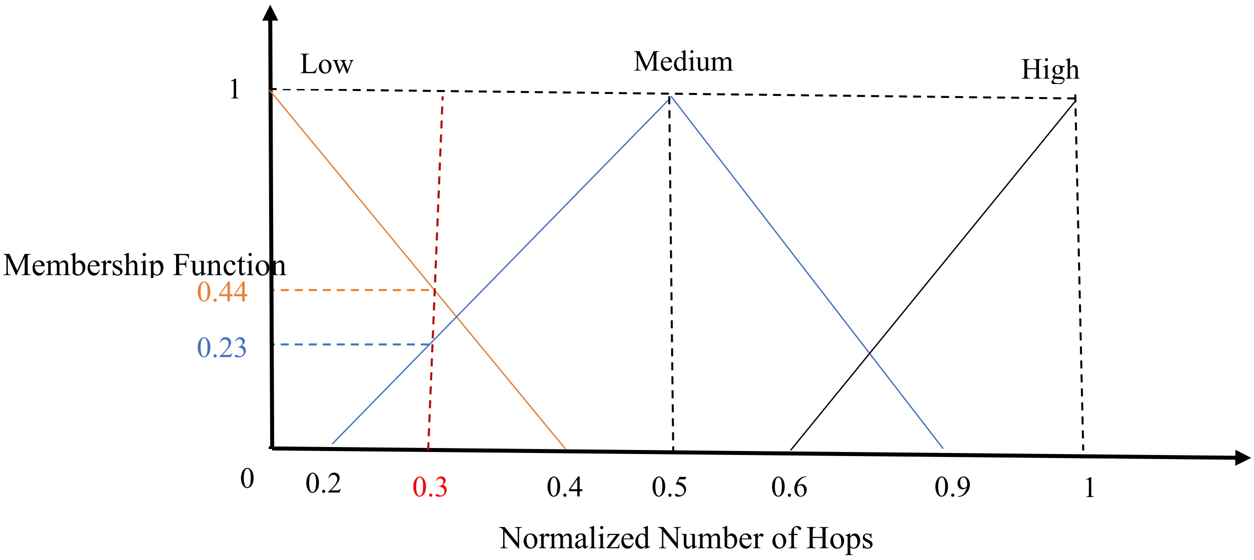


Figure 1: Normalized Number of Hops and its membership function

As indicated in Figure 1 above, the normalized number of hops hp_{nor} can be mapped to corresponding values on membership function set out in chapter 6 (Table 6-1) for the linguistic variables of *Low, Medium, and High* with values as *0.44, 023, and 0* respectively. Similarly, we can represent the latency with a route lifetime of *20ms*, with a corresponding value for its membership function described in chapter 6 (Table 6-1) as *Short, Medium, and Long* with values as *0, 0.84, and 0.15* respectively as in Figure 2. The values of *short, medium, and long* were obtained from a range of average values utilized during the research simulation.

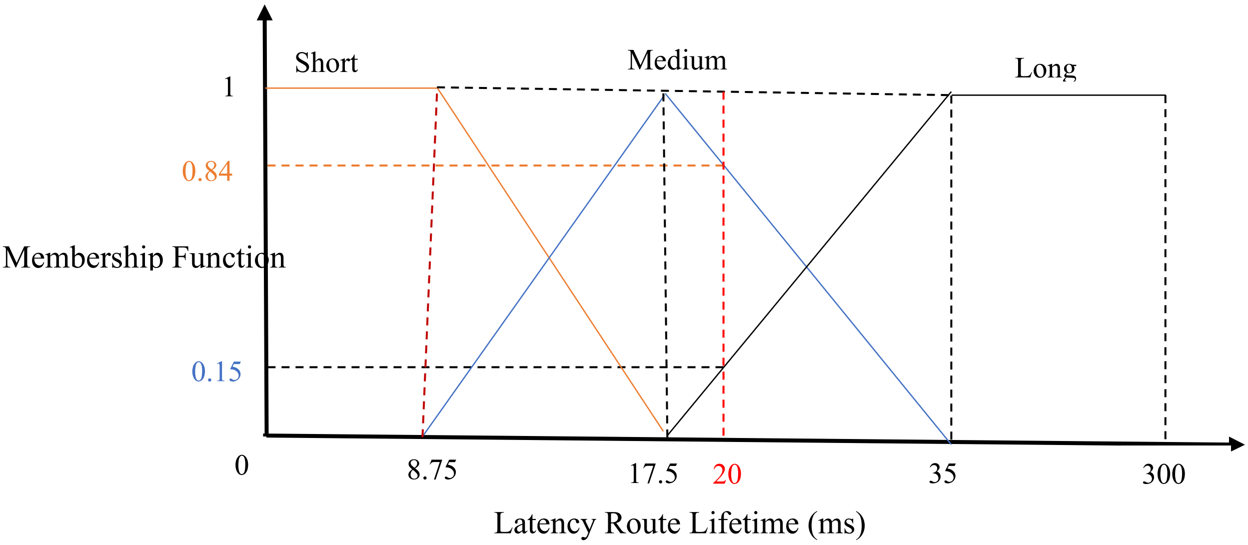


Figure 2: Latency Route Lifetime and its Membership Function

Therefore, according to the rules set by the FLS in chapter 6 while utilizing the *Max-min* method in chapter 3.9 and considering the linguistic variables defined using the *'IF-THEN',* the rules can be mapped to the different levels according to the Table 1, where the membership value of *0.44* belongs to the fuzzy set of *Low*. Where the input of the latency route lifetime has the value of *0* belonging to the fuzzy set of *short*, then *Max-min* method is executed as shown in Table 1.

Table 1: Interference rule for Latency Route Lifetime

S/N	***Hop Count***	***Latency Route Lifetime***	***Route Stability***
1.	*0.44*	*0*	*Min (0.44, 0) = 0.*
2.	*0.44*	*0.84*	*Min (0.44, 0.84) = 0.44*
3.	*0.44*	*0.15*	*Min (0.44, 0.15) = 0.15*
4.	*0.23*	*0*	*Min (0.23, 0) = 0*
5.	*0.23*	*0.84*	*Min (0.23, 0.84) = 0.23*
6.	*0.23*	*0.15*	*Min (0.23, 0.15) = 0.15*
7.	*0*	*0*	*Min (0, 0) = 0*
8.	*0*	*0.84*	*Min (0, 0.84) = 0*
9.	*0*	*0.15*	*Min (0, 0.15) = 0*

Additionally, each entry on Table 1 is considered as a rule which are classified into fuzzy sets of rules where *Low* represents *Rule 4* and *Rule 7*, while *medium Rule 1*, *Rule 5*, and *Rule 8*, and *High* represents *Rule 2*, *Rule 3*, *Rule 6*, and *Rule 9 (see Figure)*. Then by implementing *Max-min* method, the equation will be described as in Table 2,

- For *Low = max (Rule 4, Rule 7) = max (0,0) = 0*, defined as dm^l*;*
- For *Medium = max (Rule 1, Rule 5, Rule 8) = max (0, 0.23, 0) = 0.23*, defined as dm^m;
- For *High = max (Rule 2, Rule 3, Rule 6, Rule 9) = max (0.44, 0.15, 0.15, 0) = 0.44*, defined as dm^h;

Latency route lifetime as described in chapter 6 of this work has been normalized to be in the range of *[0,1]*. The centre points for *Low, Medium, and High* has been assigned variables of $CP^l, CP^m,\ and\ CP^h$, with values *0, 0.5, and 1*. These values can be defuzzified using equation in chapter 2,

$$z^* = \frac{CP^l\ .\ dm^l +\ CP^m\ .\ dm^m + CP^h\ .\ dm^h}{dm^l + dm^m + dm^h}$$

$$= \frac{0\ .\ 0 + 0.23\ .\ 0.5 + 0.44\ .\ 1}{0\ +\ 0.23\ +\ 0.44}$$

$$=\ 0.828,$$

Hence, the value *0.828ms* is the stability latency route lifetime of the fuzzification process that has been converted using a linear combination of metrics.

Appendix 8: Simulation Data Set

02:46.5	1045	44	64	0	82	0	3	1	2	0	22	10525	0	947	12943
02:53.3	1840	25	64	0	85	0	11	1	1	0	22	10970	0	430	14053
03:09.0	920	50	64	0	93	0	9	1	3	0	22	11956	0	1113	14684
03:13.7	707	65	64	0	96	0	4	1	2	0	22	12107	0	970	15028
03:39.2	851	54	64	0	108	0	6	1	2	0	22	13905	0	1213	17181
03:41.2	1179	39	64	0	109	0	2	1	2	0	22	14067	0	1185	17426
03:43.1	1840	25	64	0	110	0	10	1	1	0	22	14126	0	1019	17670
04:06.1	1840	25	64	0	122	0	10	2	1	0	22	15644	0	1036	15148
04:37.5	741	62	64	0	137	0	5	2	2	0	22	17628	0	1919	15644
05:10.2	505	91	64	0	154	0	2	2	2	0	22	19702	0	1371	13648
05:10.8	851	54	64	0	154	0	3	2	2	0	22	19757	0	1033	11270
05:27.0	505	91	64	0	162	0	9	2	2	0	22	20800	0	785	11001
05:28.7	1840	25	64	0	163	0	6	2	1	0	22	20948	0	1265	17508
05:31.5	180	255	64	0	164	0	4	2	3	0	22	20867	0	1563	21786
05:38.6	1840	25	64	0	168	0	11	2	1	0	22	21585	0	518	13629
06:13.0	180	255	64	0	185	0	4	3	3	0	22	23766	0	724	14729
06:20.2	666	69	64	0	189	0	2	3	2	0	22	24235	0	889	11193
06:27.5	851	54	64	0	192	0	3	3	2	0	22	24547	0	691	12076
06:51.1	1840	25	64	0	204	0	10	3	1	0	22	26199	0	516	13550
06:53.6	1840	25	64	0	206	0	11	3	1	0	22	26383	0	373	12416
06:54.5	528	87	64	0	206	0	9	3	2	0	22	26394	0	1094	13817
07:14.7	460	100	64	0	216	0	5	3	2	0	22	27689	0	918	12490
07:52.9	1840	25	64	0	235	0	6	3	1	0	22	30175	0	830	11467
08:34.8	1840	25	64	0	256	0	11	4	1	0	22	32864	0	386	16889
08:47.0	793	58	64	0	262	0	9	4	2	0	22	33597	0	539	18660
08:53.8	1840	25	64	0	266	0	10	4	1	0	22	34053	0	854	20080
09:13.2	638	72	64	0	275	0	2	4	2	0	22	35309	0	478	14281

09:13.9	1840	25	64	0	276	0	6	4	1	0	22	35360	0	901	12920
09:15.0	686	67	64	0	276	0	3	4	2	0	22	35375	0	712	13718
10:00.2	938	49	64	0	299	0	5	4	2	0	22	38224	0	671	13368
10:12.5	1840	25	64	0	305	0	6	5	1	0	22	39112	0	1187	18816
10:23.2	958	48	64	0	310	0	2	5	3	0	22	39775	0	227	11677
10:44.5	1840	25	64	0	321	0	10	5	1	0	22	41140	0	459	18429
11:14.8	730	63	64	0	336	0	3	5	2	0	22	43049	0	886	19567
11:26.4	779	59	64	0	342	0	5	5	2	0	22	43753	0	659	14077
11:41.9	1840	25	64	0	350	0	11	5	1	0	22	44834	0	398	15555
12:03.5	621	74	64	0	360	0	9	5	2	0	22	46113	0	429	16251
12:23.5	766	60	64	0	370	0	9	6	2	0	22	47457	0	728	13602
12:30.7	766	60	64	0	374	0	2	6	2	0	22	47949	0	1016	20773
12:41.7	754	61	64	0	380	0	5	6	2	0	22	48625	0	559	12426
12:42.3	1840	25	64	0	380	0	6	6	1	0	22	48696	0	858	11914
12:50.7	1840	25	64	0	384	0	11	6	1	0	22	49239	0	306	11435
13:14.3	1840	25	64	0	396	0	10	6	1	0	22	50720	0	423	12343
13:43.0	821	56	64	0	410	0	3	6	2	0	22	52529	0	472	12162
14:22.8	1840	25	64	0	430	0	11	7	1	0	22	55137	0	162	15559
14:39.2	938	49	64	0	438	0	10	7	2	0	22	56151	0	704	13772
14:52.0	1840	25	64	0	445	0	6	7	1	0	22	56999	0	1045	21085
15:33.2	978	47	64	0	465	0	9	7	2	0	22	59555	0	312	15810
15:35.2	422	109	64	0	466	0	3	7	3	0	22	59705	0	848	18279
15:45.7	978	47	64	0	472	0	5	7	2	0	22	60403	0	627	15070
15:56.7	978	47	64	0	477	0	2	7	2	0	22	61084	0	353	17153
16:19.5	1840	25	64	0	488	0	11	8	1	0	22	62602	0	408	19490
16:24.1	807	57	64	0	491	0	3	8	2	0	22	62837	0	775	15920
17:04.2	1840	25	64	0	511	0	10	8	1	0	22	65438	0	514	11862
17:05.8	1840	25	64	0	512	0	6	8	1	0	22	26	0	1151	21673
17:32.2	884	52	64	0	525	0	9	8	2	0	22	1678	0	345	20069
17:51.5	884	52	64	0	534	0	2	8	2	0	22	2942	0	527	19182

17:55.5	884	52	64	0	536	0	5	8	2	0	22	3172	0	939	21198
18:17.0	867	53	64	0	547	0	2	9	2	0	22	4565	0	364	16942
18:28.0	1840	25	64	0	553	0	6	9	1	0	22	5266	0	852	13114
18:50.2	621	74	64	0	564	0	5	9	2	0	22	6674	0	847	17821
19:00.5	1840	25	64	0	569	0	10	9	1	0	22	7346	0	473	19368
19:26.3	1840	25	64	0	582	0	11	9	1	0	22	9024	0	338	15599
19:37.5	534	86	64	0	587	0	9	9	2	0	22	9696	0	259	21112
19:48.5	1000	46	64	0	593	0	3	9	2	0	22	10389	0	694	16748
20:04.2	500	92	64	0	601	0	2	10	2	0	22	11437	0	310	18007
20:06.5	500	92	64	0	602	0	9	10	2	0	22	11544	0	260	19439
20:20.7	500	92	64	0	609	0	5	10	2	0	22	12469	0	500	14947
20:52.3	1840	25	64	0	625	0	10	10	1	0	22	14503	0	438	18636
21:12.8	1840	25	64	0	635	0	11	10	1	0	22	15837	0	527	17633
21:16.0	779	59	64	0	637	0	3	10	2	0	22	15901	0	639	14053
21:27.3	1840	25	64	0	642	0	6	10	1	0	22	16764	0	658	14665
22:21.7	489	94	64	0	670	0	2	11	2	0	22	20231	0	358	11362
22:26.0	489	94	64	0	672	0	9	11	2	0	22	20477	0	214	11691
22:48.8	779	59	64	0	683	0	3	11	2	0	22	21930	0	572	15498
22:50.0	489	94	64	0	684	0	5	11	2	0	22	22015	0	581	12141
23:08.4	1840	25	64	0	693	0	10	11	1	0	22	23202	0	216	11376
23:29.3	1840	25	64	0	703	0	6	11	1	0	22	24568	0	580	20221
23:45.0	1840	25	64	0	711	0	11	11	1	0	22	25576	0	119	12861
24:15.2	489	94	64	0	726	0	9	12	2	0	22	27476	0	169	18487
24:45.5	270	170	64	0	742	0	3	12	2	0	22	29315	0	683	19000
24:45.9	1840	25	64	0	742	0	11	12	1	0	22	29476	0	240	20551
24:59.5	489	94	64	0	748	0	2	12	2	0	22	30341	0	217	13257
25:14.2	1840	25	64	0	756	0	6	12	1	0	22	31284	0	318	17584
25:38.0	489	94	64	0	768	0	5	12	2	0	22	32768	0	446	13885
25:52.1	1840	25	64	0	775	0	10	12	1	0	22	33662	0	226	13713
26:17.2	489	94	64	0	787	0	9	13	2	0	22	35285	0	303	20508

26:19.0	1840	25	64	0	788	0	11	13	1	0	22	35434	0	109	15773
26:37.7	1840	25	64	0	798	0	10	13	1	0	22	36611	0	223	15496
27:25.0	1840	25	64	0	821	0	6	13	1	0	22	39657	0	608	21711
27:27.5	489	94	64	0	823	0	2	13	3	0	22	39804	0	153	12459
27:43.7	766	60	64	0	831	0	5	13	2	0	22	40821	0	829	20637
27:45.8	793	58	64	0	832	0	3	13	2	0	22	40884	0	537	14881
28:10.5	597	77	64	0	844	0	5	14	2	0	22	42520	0	714	17399
28:16.7	597	77	64	0	847	0	2	14	2	0	22	42955	0	651	16147
28:28.2	741	62	64	0	853	0	9	14	2	0	22	43666	0	132	11035
28:41.6	1840	25	64	0	860	0	6	14	1	0	22	44557	0	733	12328
29:22.9	1840	25	64	0	880	0	11	14	1	0	22	47205	0	139	15549
29:34.6	1840	25	64	0	886	0	10	14	1	0	22	47929	0	431	14652
29:49.3	821	56	64	0	893	0	3	14	2	0	22	48838	0	813	20389
30:04.7	647	71	64	0	901	0	9	15	2	0	22	49837	0	145	16300
30:38.3	605	76	64	0	918	0	3	15	2	0	22	51972	0	912	15795
30:58.7	779	59	64	0	928	0	5	15	2	0	22	53299	0	576	13788
31:01.6	1840	25	64	0	930	0	11	15	1	0	22	53503	0	337	16450
31:27.6	1840	25	64	0	943	0	10	15	1	0	22	55164	0	912	18372
31:46.7	851	54	64	0	952	0	2	15	2	0	22	56375	0	459	17427
31:54.9	1840	25	64	0	956	0	6	15	1	0	22	56926	0	816	15668
32:28.5	1840	25	64	0	973	0	11	16	1	0	22	59081	0	196	14672
32:31.9	1840	25	64	0	975	0	6	16	1	0	22	59297	0	326	12314
32:52.0	867	53	64	0	985	0	9	16	2	0	22	60546	0	368	13902
33:30.2	901	51	64	0	1004	0	2	16	2	0	22	62971	0	388	17192
33:34.8	920	50	64	0	1006	0	3	16	2	0	22	63276	0	632	14432
33:50.2	696	66	64	0	1014	0	5	16	2	0	22	64229	0	758	13809
33:53.4	1840	25	64	0	1015	0	10	16	1	0	22	64490	0	553	11876
34:17.5	920	50	64	0	1027	0	9	17	2	0	22	459	0	251	14271
34:27.0	884	52	64	0	1032	0	5	17	2	0	22	1084	0	504	12241
34:38.5	1840	25	64	0	1038	0	6	17	1	0	22	1861	0	1262	20328

34:41.1	1840	25	64	0	1039	0	10	17	1	0	22	2010	0	683	15608
34:42.2	821	56	64	0	1040	0	2	17	2	0	22	2033	0	494	11762
35:07.7	77	590	64	0	1053	0	4	17	3	0	22	3700	0	350	18352
35:41.5	978	47	64	0	1070	0	3	17	2	0	22	5845	0	784	20821
35:42.3	1840	25	64	0	1070	0	11	17	1	0	22	5911	0	273	16207