BACHELOR THESIS

Thuy An Nguyen

Zielfunktionen für das Routing-Protokoll RPL: Erweiterungen und experimentelle Analyse mit dem IoT Betriebssystem RIOT

Faculty of Computer Science and Digital Society

HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Thuy An Nguyen

Zielfunktionen für das Routing-Protokoll RPL: Erweiterungen und experimentelle Analyse mit dem IoT Betriebssystem RIOT

Bachelor thesis submitted for examination in Bachelor´s degree
in the study course *Bachelor of Science Informatik Technischer Systeme*
at the Faculty of Computer Science and Digital Society
at University of Applied Sciences Hamburg

Supervisor: Prof. Dr. Thomas Schmidt
Supervisor: Prof. Dr. Franz Korf

Submitted on: 05. August 2026

Thuy An Nguyen

Thema der Arbeit

Zielfunktionen für das Routing-Protokoll RPL: Erweiterungen und experimentelle Analyse mit dem IoT Betriebssystem RIOT

Stichwörter

IoT, RPL, OF, Zielfunktion, RIOT, Sensorsysteme, Netzwerk, Kommunikationsprotokoll, LLN, OF0, MRHOF

Kurzzusammenfassung

RPL ist ein Routing Protkoll für Netzwerke, die über wenig Energie verfügen und in denen hoher Datenverlust auftritt. Eine Konfiguration von RPL ist die Zielfunktion (OF), welche bestimmt wie Knoten ihren Elternknoten auswählen und ihren Rank im Netzwerk berechnen. RIOT OS, ein Betriebssystem für eingebettete Systeme, verfügt aktuell über eine OF, OF0.
Diese Arbeit implementiert die OF MRHOF in RPL, in welche verschiedene Metriken eingesetzt werden können, um die Topologie des Netzwerk aufzubauen. Der Fokus dieser Arbeit ist die aktuelle Forschung verschiedener OFs, die Implementierung der MRHOF in RIOT OS und die Bewertung dieser OF in einem hardware Experiment.

Thuy An Nguyen

Title of Thesis

Objective functions for the RPL routing protocol: Extensions and experimental analysis with the IoT operating system RIOT

Keywords

IoT, RPL, OF, Objective Function, RIOT, sensor, system, network, communication protocol, LLN, OF0, MRHOF

Abstract

RPL is a routing protocol designed for lower power and lossy networks. One of its configurations is the Objective Function (OF) which determines how nodes choose their parent and calculate their rank within the network. RIOT OS, an operating system for embedded devices, currently implements one OF, OF0.

This thesis implements the OF MRHOF which can be employed by choosing from different routing metrics. It focuses on exploring different OFs, how to implement an OF in RIOT OS and evaluating its efficiency in a real world experiment using different network performance metrics.

Acknowledgements

An dieser Stelle mit ich mich bei all denen bedanken, die mich während meines Studiums und insbesondere bei der Erarbeitung dieser Abschlussarbeit unterstützt haben.

Ganz herzlich möchte mich bei meinem Betreuer Prof. Dr. Thomas Schmidt und Elena Frank bedanken, die mich während des gesamten Prozess meiner Bachelorarbeit, von der Themenfindung, bis zum Planen, Implementieren und Experimentieren sowohl wie beim Schreiben unterstützt haben. Ebenso danke ich Prof. Dr. Franz Korf, dass er sich für die Korrektur dieser Arbeit zur Verfügung gestellt hat.

Besonders Elena möchte ich ein großes Dankeschön aussprechen, die mir jede noch so kleinste Frage geduldig beantwortet hat und zu jedem Zeitpunkt hilfsbereit zur Seite stand.

Auch möchte ich mich bei meiner Familie und meinen Freunden bedanken, die fleißig Korrektur gelesen haben und mich im Laufe meines Studiums motiviert und ermutigt haben.

Contents

List of Figures

List of Tables

Abbreviations

ADC Analog-to-Digital Converter.

CBR Constant Bit Rate.

CDF Cumulative Distribution Function.

DAG Directed Acyclic Graph.

DAO Destination Advertisement Object.

DAO-ACK Destination Advertisement Object Acknowledgement.

DIO DODAG Information Object.

DIS DODAG Information Solicitation.

DODAG Destination-Oriented Directed Acyclic Graph.

ELT Expected Lifetime.

ETX Expected Transmission Count.

HC Hop Count.

IEEE Institute of Electrical and Electronics Engineers.

IETF Internet Engineering Task Force.

LLNs low power and lossy networks.

LQI Link Quality Index.

MOP Mode of Operation.

MP2P Multi-Point to Point.

MRHOF Minimum Rank with Hysteresis Objective Function.

netstats Network statistics.

OCP Objective Code Point.

OF Objective Function.

OF0 Objective Function 0.

P2MP Point to Multi-Point.

P2P Point to Point.

PDR Packet Delivery Ratio.

PLR Packet Loss Ratio.

PR Pull Request.

ROLL Routing over Low power and Lossy networks.

RPL Routing Protocol for Low-Power and Lossy Networks.

RTT Round Trip Time.

1 Introduction

1.1 Motivation

IoT multihop networks are used to collect and process data in areas where devices cannot connect directly to a central base station. Instead of each individual device trying to reach a faraway base station on its own, the devices work together to form a network and transfer data from one neighbor to the next in order to transport packets beyond the communication range of each device.

Such networks are used in application areas such as Industrial IoT or Smart Agriculture to operate automated factories or manage large-scale food production but also in everyday use such as Smart Cities to manage urban infrastructure like public transit or traffic lights.

As these network devices are spread across large areas, they often contend with limited power, memory, and processing resources. Therefore, routes to transfer information between the devices must be chosen carefully in order not to drain these scarce resources quickly. Routing Protocol for Low-Power and Lossy Networks (RPL) [2] as a standard routing protocol was specifically designed for low-power, low-cost, and resource-constrained networks. The RPL protocol constructs a tree-like topology of these devices, from here on called nodes, to determine appropriate routing paths for the nodes within a network.

To deal with the various requirements of different networks, one configurable parameter of RPL is the Objective Function (OF), which is used to adapt a network topology to its intended goals and accommodate each of these unique use cases. Different OFs pursue different goals and operate under different constraints. Our goal is to research the current state of OFs and compare what impact they have on a network.

1.2 Approach and Goal

This thesis and project will study the characteristics and differences of standardized OFs in order to discern their impact on the behaviour and performance of a network when RPL is used as the routing protocol. This research hence contains an implementation of an OF for RIOT OS [3] an open-source operating system for IoT and other embedded devices and an experimental analysis, conducted using the FIT IoT-LAB [1].

1.3 Thesis Structure

First, we will provide background information about IoT multihop networks, the RPL protocol, RIOT OS [3] and the boards employed for the experiments in chapter 2. In chapter 3 we will outline the problem focus of this thesis and issues with the current OF implementation of RIOT OS [3] which we hope to improve. In chapter 4, we will implement the changes to address the issues in chapter 3. Chapter 5 will be an experimental analysis, from planning and conducting the experiment to the interpretation of the results. Finally, we will summarize the results and findings of this work.

2 Background

2.1 IoT Multihop Networks

In many IoT networks data travels through more than one wireless device to its destination. The nearest reachable node is usually not the final destination. It is just one step toward the destination, which is also called a hop. As data has to travel through multiple nodes to reach its destination, this kind of network is called a multihop network.

Oftentimes the devices used in these kind of networks are constrained nodes. Constrained nodes lack certain resources that are often expected and desired in network nodes. Constrained nodes are commonly limited in power, memory, processing resources, code space, and processing cycles. The reason for these constraints are usually cost or physical limitations such as size, weight or available energy [4].

Networks made up of those constrained nodes are called low power and lossy networks (LLNs) [4]. To combat the LLNs challenges of considerable loss at the physical layer, significant variability of the delivery rate, short-term unreliability and some instability, many different routing protocols have been proprosed or adapted for use in LLNs.

Some examples are AODV [5], LOAD [6], ZigBee Cluster Tree [7], CTP [8] and RPL [2]. However, each of these protocols have notable limitations. RPL is highly flexible and dynamic. It supports both hierarchical and flat topologies wheras many others only support one of the two. While RPL can handle different types of traffic, Multi-Point to Point (MP2P), Point to Multi-Point (P2MP), and Point to Point (P2P), none of the mentioned routing protocols support all of three these traffic types.

One routing protocol that does fulfill these conditions, as RPL does, is Hydro [9]. Its main difference to RPL is that routes are discovered on-demand, meaning when they are needed, while in RPL routes are pre-computed and maintained.

2.2 RPL Protocol Overview

RPL is a routing protocol especially designed for LLNs. Its topology is based on a Directed Acyclic Graph (DAG) [2]. A DAG is a graph consisting of nodes connected by directed edges, where the connections move strictly in a single direction and never return to the same node to form a closed cycle. Paths within a DAG always terminate at a root node, which is a node which has no outgoing edges.

In a DAG each node can have multiple parents and has at least one parent to whom they route their own and packets from other nodes. That node is referred to as the preferred parent. The only nodes that do not have any parent are root nodes as a path terminates there.

The clear direction of the graph and its acyclic attribute offers many adavantages. It prevents the network from unnecessarily transmitting the same data to the same nodes more than once and thus using more energy and bandwidth than needed. It also prevents routing loops where a data packet is continously forwarded among a closed loop of nodes without ever reaching its final destination.

One DAG is contained in a RPL instance. A physical network of nodes can run multiple RPL instances concurrently to serve different constraints. Each of these RPL instances is identifiable by a unique RPL instance ID. RPL partitions a DAG into several Destination-Oriented Directed Acyclic Graph (DODAG). Different from a DAG, each DODAG only has one root node. The root may also act as a border router, aggregate routes in the DODAG and redistribute DODAG routes into other routing protocols.

Although a node can be part of several RPL instances, it can only join exactly one DODAG inside that RPL instance.

For each DODAG root a goal is set, such as to construct a DODAG using a specific OF while maintaining a reliable connection to a specific database. The goal is usually set by a network operator or a developer. The DODAG root is responsible for achieving this goal and knowing whether it is being fulfilled or not. If the goal is fulfilled, this DODAG is grounded. A DODAG that does not fulfill the set goal is floating.

Construction of Topology In order to establish a DODAG, a node first has to declare itself as a DODAG root. The DODAG of the root has a DODAG version number that is usually incremented when global repairs are performed for the DODAG. Any node, that joins the DODAG, chooses a node inside the DODAG to which it will route its packets that are going in the direction of the root. The direct receiver of the packets is called a parent of the sender node.

Traffic traveling towards the root is called upward traffic. Traffic traveling away from the root is called downward traffic. Downward routing inside a network is defined by the Mode of Operation (MOP) of the DODAG. A DODAG can either operate in storing mode or non-storing mode.

There are three types of nodes within a DODAG:

1. The **DODAG root** has the ability to construct a DODAG and is also a gateway to other networks. In non-storing mode, the DODAG root is the only node in the network that has to store and maintain downward routing tables where the routes to reach nodes within the DODAG are saved.

2. The **routers**, which are parents to other nodes, collect, generate and forward traffic within the network. If the network is operating in storing mode, routers have to maintain routing table entries for all nodes whose packets would pass them on the way to the root alongside the root. In non-storing mode, the root bears this responsibility by itself.

3. The **leaves** join an existing DODAG and generate traffic but do not forward other nodes' packets.

RPL supports three traffic patterns:

1. **MP2P** is upward routing where data is sent from multiple nodes towards a DODAG root. Upward routing operates identically in both storing mode and non-storing mode.

2. **P2MP** is downward routing where data is sent from the DODAG root to multiple other nodes. In storing mode, routers look up the next router to forward the packet to in their saved routing tables. In non-storing mode, the root calculates the entire path beforehand using its routing tables and attaches a source routing header to the packet. This source routing header tells the router nodes where to forward a packet to next.

3. **P2P** is the pattern where traffic is forwarded between two nodes. In non-storing node a packet has to travel all the way up to the DODAG root before it can travel back down to its destination as only the root maintains all routing table entries. In storing mode a packet only travels until it reaches a node on the path to the root of both the sender and receiver node. This common node on both paths then immediately redirects the packet downward.

The process of joining the DODAG consists of the joining node and their prospective parent exchanging control messages. Based on these control messages, nodes decide whether to join a DODAG or not and which node to choose as their parent. All control messages are described below and Fig. 2.1 showcases in which order the control messages are sent.

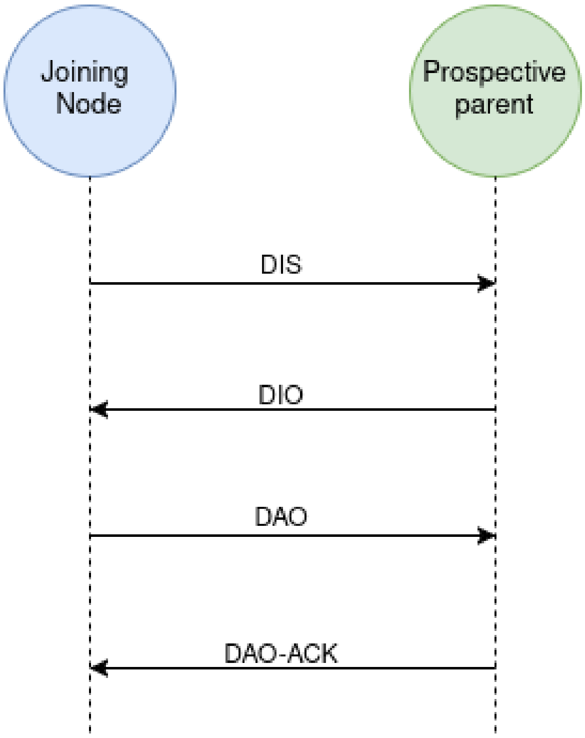


Figure 2.1: Sequence diagram of control messages during topology construction

- **DODAG Information Object (DIO)** are broadcast downwards by the DODAG root and parent routers to their neighboring nodes, so a node that has yet to join an existing RPL Instance can receive information about it.

- **DODAG Information Solicitation (DIS)** is sent from a node outside the DODAG to its neighboring nodes, which are routers of a DODAG. It is used to proactively request a DIO message from nearby RPL routers.
- **Destination Advertisement Object (DAO)** is sent from a child node upward to its parent(s) or the DODAG root. The receiver of the DAO message uses that information to establish and maintain downward routes.
- **Destination Advertisement Object Acknowledgement (DAO-ACK)** is sent by the receiver of a DAO message to confirm its reception.

Objective Function Upon joining a DODAG, a node will calculate its rank inside this DODAG. The rank defines the individual position of a node relative to other nodes with respect to the DODAG root and is determined by the RPL network's OF, which is the focus of this thesis. Nodes farther away from the root have a higher rank, and nodes near the root have a lower rank. Using the OF, the node also decides how parents in the DODAG are selected and therefore, how the DODAG is formed. The two standardized OFs are OF0 [10] and MRHOF [11].

DODAG Repair Global repairs are initiated by the DODAG root when the root detects a failure or inefficiency. They can also be scheduled in advance or be initiated due to administrative intervention. When a new DODAG Version number is introduced, nodes are released from their current rank and can select a new position and a new parent.

Besides global repairs, local repairs may be carried out by individual nodes. For example, these nodes may experience local link failure and thus notify their children to find alternative parents.

Security There are three RPL security modes.

- **Unsecured Mode** transmits RPL control messages without any additional protocol-level security. This mode often relies on other security measures at other network layers.

- **Preinstalled Mode** where joining nodes use pre-configured symmetric keys to process and generate RPL control messages. This provides message confidentiality, integrity and authenticity.
- **Authenticated Mode** where nodes possess a preinstalled key that allows them to join a secure RPL instance only as a leaf node. To join as a router, the node must dynamically authenticate with a key authority to obtain a second, more privileged key.

2.3 RIOT OS

For research and experiments we are using RIOT OS [3], an open-source operating system for IoT devices. RIOT OS [3] offers different modules to be incorporated into and used inside a project. Through its modular structure needed modules can be added and removed at will to extend the program capabilities or downsize the application to its needs swiftly.

For example, experiments conducted for this thesis made use of modules already available in RIOT OS [3].

- UDP sock API to send and receive data packets on the transport layer
- Network statistics (netstats) neighbor to access packet statistics for RPL
- Generic (GNRC) network stack as the RIOT OS [3] default network stack which includes IPv6 and 6LoWPAN, the network and adaptation layer to RPL [2]
- Peripheral Driver Interface to communicate with external hardware
- RPL

RIOT OS [3] can be employed on 250 different boards. The following section will list some examples which have been used in experiments and tests for this thesis.

2.4 Employed Boards

In our experiments, we used three kinds of hardware boards. These boards all support packet transmission via IEEE 802.15.4 but were chosen for testing and experiments for their differing functions.

2.4.1 Adafruit Feather nRF52840 Sense

The Adafruit Feather nRF52840 Sense [12] is a development board usually used for environmental monitoring, wearable technology, gesture-controlled projects and more. It possesses a 32-bit ARM Cortex M4F processor as well as many different sensors, such as a gyroscope, a magnometer, a humidity sensor, temperature sensor, etc. As this board is used and available at the University of Applied Sciences Hamburg for projects, we chose this board for initial testing and also later for local experiments.

2.4.2 nRF52840dk

The Nordic nRF52840 DK [13] is a development board used for prototyping and testing. It possesses a 32-bit ARM Cortex M4F processor and offers hardware controls such as buttons and LEDs. This board can be controlled remotely on the FIT IoT-LAB [1] remote testbed we were using to conduct experiments and was therefore easier to use for multiple repeated experiments.

2.4.3 M3

The IoT-LAB M3 board [14] is based on an STM32 (ARM Cortex M3) micro-controller, designed specifically for the FIT IoT-LAB [1] testbed. It is primarily used for large-scale wireless sensor network experimentation. It offers a variety of sensors, such as a temperature sensor, a pressure sensor, and a gyroscope. It is by far the most employed board on the FIT IoT-LAB [1].

2.5 Network Comparison Methods

In order to compare OFs with each other, there are different network performance metrics which can be recorded during experiments. In order to have a common understanding of some commonly used ones, we will outline them in Table 2.1.

Table 2.1: Network comparison metrics

Metric	Unit	Description	Comparison
PDR	%	Ratio of successfully delivered packets to total sent packets	High PDR indicates reliable network. Low PDR may indicate congestion, interference or routing failures.
Packet Loss Ratio (PLR)	%	Percentage of packets that fail to reach their destination	Complementary to PDR. Low PLR indicates a reliable network.
Energy consumption	kWh	Total or average energy used by nodes	Low energy consumption indicates a more efficient network.
Network lifetime	Time unit, such as seconds, minutes, hours or days	Time until first node depletes its energy	Longer network lifetime indicates a more efficient network.
End-to-end delay	Time unit, such as ms, s or µ	Total time for a packet to travel from its source to its destination	Lower end-to-end delay indicates a more efficient network.
RTT	Time unit, such as ms, s or µs	Time for a packet to travel to its destination and back	Lower end-to-end delay indicates a more efficient network.
Throughput	bps (byte per seconds) or kbps (kilobyte per second)	Rate of successful data delivery over the network.	High throughput indicates a reliable network. Low throughput may indicate congestion.
Overhead	% or bytes	Amount of non-data traffic (e.g. control messages, headers, acknowledgement messages) relative to actual data traffic	Low overhead indicates a more efficient network.

3 Related Work and Problem Analysis

3.1 Related Works

This chapter reviews existing work relevant to our RPL OF implementation. It examines prior efforts in comparing MRHOF to OF0, the two OFs standardized by the Internet Engineering Task Force (IETF), and other works researching energy-based OFs.

3.1.1 Standard Objective Function Comparison: OF0 and MRHOF

The conference paper "A comparative study of RPL objective functions"[15] by Kechiche et al. researched OF0 and MRHOF in an experiment with a varied number of nodes, ranging from 20 to 140 in intervals of 20, employing exactly one root node. The nodes are always on the same area of deployment of $300 \times 300\ m_2$, meaning a higher number of nodes indicates a higher network density. MRHOF was employed once using ETX as the routing metric and once using remaining energy. The experiment was run on the Contiki OS [16] making use of its Cooja [17] simulator. The simulation lasted 1000 seconds each time with every node sending one packet every four minutes, after experimenting with packet loads of one packet every one to six seconds and deeming the decided interval the ideal one. The node transmission power was set to 0 dBm.

The authors found MRHOF-ETX and OF0 to perform poorly in terms of PDR with a number of nodes below 60. Both OFs reach a peak PDR value around a node number of 60%. While the OF PDR only descends slowly as the amount of nodes rises, the MRHOF-ETX PDR values plummet quickly to a PDR of 30% at 140 nodes. MRHOF-Energy meanwhile only reached a peak PDR of 30% once at 80 nodes and remained a steady PDR of 10% from there on.

In terms of end-to-end delay, OF0 and MRHOF-ETX reached a value of about 300 seconds for 20 nodes. This end-to-end delay quickly sinks to values of about four seconds for both of them and remains about the same from there even for a higher node amount. MRHOF-Energy starts at a lower end-to-end delay of 225 seconds and also decreases quickly but maintains a higher value of about 40 seconds even for a large node amount.

Overall, the authors deduced OF0 and MRHOF-ETX to not support high load and MRHOF-ETX to not support network scalability as PDR values degraded significantly with a growing number of nodes. MRHOF-Energy was shown to provide poor performance overall.

Alayed et al. published their conference paper "Evaluation of RPL's Single Metric Objective Functions"[18] and conducted similar experiments. Their experiment was also conducted using the Cooja [17] simulator by Contiki OS [16]. They also used one root. The number of nodes ranged from 11 to 46 in 5 node increments on an area of 150×150 m_2. Each node sent about one packet per minute and the simulation lasted for an hour each time. In their experiments, they compared the two OFs OF0 and MRHOF-ETX.

Instead of PDR, the authors measured PLR. They observed both the OF0 and the MRHOF PLR values to increase proportionally as the number of nodes grew and the OF0 PLR values to be slightly higher than the MRHOF-ETX values by about 4% for a node amount below 30. The values of both OFs were in the range of 0% to 10% below 41 nodes. From 41 nodes and onwards, OF0 outperformed MRHOF-ETX in terms of PLR.

These findings do not align with Kechiche et al.'s experiment results where PLR was always above 90% for a node amount less than 50. It may however be due to Kechiche et al.'s experiment nodes being spread on an area double the size of the one used in Alayed et al.'s experiment. If the nodes were distributed in even spacing on the experiment area, some nodes may have been too far to connect to the root node or its children.

Gupta et al.'s more recent evaluation of their simulation[19] similarly observed better PDR results for MRHOF-ETX than OF0 for a node amount of 30 by about 5%. When tested on a node amount of 30 nodes, the two OFs achieved about the same PDR values. These observations support Alayed et al.'s observation.

However, a new hybrid OF, proposed by Gupta et al, outperformed both MRHOF-ETX and OF0. This OFwas called MRHOF-ETX+Energy, combining both link and energy constraints into one metric. Gupta et al. additionally examined route churn and inter-packet timing stability, offering a complementary perspective on topology behavior beyond raw delivery performance.

Alayed et al.[18] further observed a steady ascent for average end-to-end latency and a lower value for MRHOF-ETX for a node amount below 36. For node numbers higher than that, both OFs assumed the same values and rose at the same degree.

The authors also observed average power consumption for OF0 and MRHOF-ETX to be identical for a node number of 11 but the MRHOF-ETX power consumption to rise significantly as the number of nodes increased while OF0 remains steady after a small increase.

3.1.2 Real-World Testbed Evaluations

It should be noted that 86% of RPL OF evaluations strictly used simulations and 14% actual physical hardware experiments[20]. While simulations are easier to execute and can run multiple experiments with large numbers of nodes with less effort, they do not fully mirror real life experiments.

Some studies have consequently begun validating simulation results against physical hardware experiment results. For instance, Hakeem et al. ran both Cooja [17] simulations as well as experiments on the FIT IoT-LAB [1] remote testbed in their work "RPL Routing Protocol Performance in Smart Grid Applications Based Wireless Sensors: Experimental and Simulated Analysis"[21]. Once again OF0 and MRHOF-ETX were evaluated. First, in an experiment conducted at the Lille site of the FIT IoT-LAB [1]. They ran experiments on m3 nodes, flashing them using Contiki OS [16]. Within an area of about 225 m_2, 20, 40, 60, 80, 100, 120 and 140 nodes were used for experiments respectively. The nodes were always arranged in a random topology. Afterwards a simulation using Cooja [17] was run. 200, 300, 400, 500, 600, 700, 800, 900 and 1000 nodes respectively were used for experiments and arranged in a random topology as well.

The authors observed a decrease in the amount of generated control messages as the number of nodes decreased. They also observed OF0 to generate more control messages than MRHOF-ETX in both the testbed experiment and the simulation. PDR values similarly sunk as the network density increased. MRHOF-ETX also achieved higher PDR values than OF0 both in the experiment and the simulation. And even when it came to power consumption, MRHOF-ETX consumed less power than OF0 in both the experiment and simulation.

The authors also observed an increase in average latency with a higher number of nodes. MRHOF-ETX measured a lower average latency than OF0 in the testbed experiment. In the simulation average latency was similar for both OFs when 200 or 300 nodes were part of the network. When more than 300 nodes were used for the experiment, OF0 achieved a lower average latency as the number of nodes increased with the gap to MRHOF-ETX's latency widening further.

Altogether in this study MRHOF-ETX was proven to outperform OF0, especially in terms of PDR and power consumption. The authors recommeneded OF0 only to be chosen in a high density network with low latency requirements like medical monitoring applications.

However, an earlier fully real-world deployment was done by Todoli-Ferrandis et al. in their study titled "RPL Routing in a real life scenario with an energy efficient objective function. [22]. They conducted an experiment using 8 Zolertia Z1 boards, including one root node, in their office space that ran Contiki OS [16]. The authors tested OF0, MRHOF-ETX and a newly implemented OF based on remaining energy. Two kinds of experiments were conducted. One where the experiment area was 17 × 34 m_2 and all nodes could see the root node directly. And one where the experiment area was 45 × 34 m_2 and some nodes could not see the root node directly, forcing them to join the network with a node other than the root as their parent.

The experiment revealed the longest network lifetime to be achieved by the energy OF in both experiments. The authors also observed different patterns for both experiments. The smaller area experiment considered network lifetime until the last node has fully exhausted its battery. The larger area experiment considered network lifetime until all nodes that reach the root node in one hop have exhausted their battery. Even when other nodes still have energy left, they are considered dead as well because their connection to the root has died.

The authors also noted an inconsistency of the energy OF achieving a shorter delay than the expected OF0 in the smaller area experiment. In the larger area one, OF0 achieved the shortest delay out of the three OFs.

When it came to PLR, it was MRHOF-ETX that managed the smallest loss rate in both experiments.

In the end the authors concluded an OF based on optimizing energy usage to achieve an extended lifetime duration of about 30-40% but admitted that delay and loss would have to worsen in return.

Spoorthi P. Shetty and Udaya Kumar K. Shenoy published the results of their experiment on the FIT IoT-LAB [1] testbed in their research titled "Performance of RPL Objective Functions Using FIT IoT Lab"[23]. In their work, they tested the two OFs OF0 and MRHOF-ETX on the testbed in Lille using 10, 20, 30 and 40 nodes respectively. Contiki OS [16] was used to flash m3 boards. The nodes were arranged in a random topology. Both networks with one root and with multiple roots were tested.

Their experiment results indicated that MRHOF-ETX had better power consumption values for a node amount of 20 or 10 while a node amount of 30 or 40, power consumption was similar for both OFs.

The authors also tested using only one root node against using multiple root nodes. In those experiments they observed MRHOF-ETX to consume less power when there was more than one root. This observations was also their final conclusion, putting MRHOF-ETX in a slightly better light.

Notably, a later study by the same lead author Spoorthi B. Shetty (in previous research Spoorthi P. Shetty) and co-authored by Mangala Shetty "Energy Efficient RPL Objective Function Using FIT IoT-Lab"[24] conducted an experiment testing those same two OFs OF0 and MRHOF-ETX. The experiment was performed again on the FIT IoT-LAB [1], this time on the Grenoble testbed site. The nodes employed were once again m3 nodes and they were flashed using Contiki OS [16]. The experiment used 10-50 nodes, but only one root was employed this time. The authors compared a ring topology with a random topology.

They observed the power consumption to be similar for both OFs for a random topology. For a ring topology, OF0 was found to consume less power. Thus the authors concluded that OF0 should be preferred over MRHOF-ETX to achieve energy efficiency.

The results of both papers co-authored by Spoorthi P. Shetty show that many factors have an impact on the performance of an OF as in the Spoorthi P. Shetty and Udaya Kumar K. Shenoy research MRHOF-ETX was shown to be more energy efficient while in the P. Shetty and Mangala Shetty research an advantage in energy efficiency when choosing OF0 was concluded.

3.1.3 Energy-Aware Objective Functions

As studies not only focused on the OF network performance anymore but also on network lifetime, many studies now shifted their focus to implementing more energy efficient OFs as already demonstrated by Todoli-Ferrandis et al[22].

Kamgueu et al. implemented and tested a custom energy-based OF in their work "Energy-based routing metric for RPL"[25]. This custom energy-based OF uses an online real time battery level estimation model. Their battery model used current energy consumption and elapsed time to estimate battery remaining energy. The custom energy OF was implemented in Contiki OS [16]. The Cooja [17] simulator was then used to compare this custom energy OF with MRHOF-ETX. The simulation made use of 20 nodes, including one root node, on a 300×300 m_2 area, with non-root nodes sending 87 byte packets to the root node. For comparison, one experiment had the nodes send 1 packet every minute and the other 6 packets every minute.

The authors observed the custom energy OF to achieve a more even remaining energy distribution between different nodes in the network. At the end of the simulation, where nodes sent 6 packets per minute, when using the custom energy OF, more than half of the nodes had 32% remaining energy, which was also the lowest remaining energy any node had. The nodes with the most remaining energy had 48% left. Meanwhile, when using MRHOF-ETX, the remaining energy of all nodes was more uneven, with 45% of the nodes having more than 40% remaining energy, 10% of nodes having less than 25% remaining energy and the remaining 30% of nodes scattered in the middle of these values. As such, the computed network lifetime of a network using MRHOF-ETX would be 35 days, while a network using the custom energy OF would achieve a longer lifetime of 40 days.

Additionally, PDR of the two OFs was compared. The custom energy OF had a PDR of 96.6% when one packet per minute was transmitted and 94.7% when six packets per minute were transmitted. However, MRHOF-ETX always achieved a higher PDR. The higher the load was on the network, with more packets being sent per minute, the lower was the loss of MRHOF-ETX when compared to the custom energy OF.

Consequently, the research shows that the custom energy OF achieved a higher network lifetime at the cost of some transmission accuracy.

Instead of only seeking to improve energy efficiency, Chang et al. tried to combine energy awareness with link quality considerations in their work "Energy-Efficient Orineted Routing Algorithm in Wireless Sensor Networks"[26]. This is achieved through the sum of an ETX value multiplied by a scalar and an energy value multiplied by a scalar. Their OF thus combined two metrics together into a routing score called R. The OF then seeks to minimize R and in this way choose a parent with high remaining energy as well as a strong physical link.

Their OF was implemented in Contiki OS [16]. The Cooja [17] simulator was used to compare the newly proprosed OF with MRHOF-ETX. In one simulation four nodes, including the root node, were arranged in a rectangle shape with each of the fours nodes in one corner. In a second simulation two more nodes were added near the corner of the rectangle furthest away from the root node. The simulation time was set to 60 minutes.

The authors observed their OF to balance power consumption more evenly between each of the nodes when compared to MRHOF-ETX. This energy imbalance also led to one of the experiment nodes being disabled, meaning depleting all its energy, before the simulation time of 60 minutes was over when using MRHOF-ETX. The network which used the new OF survived until the end of the simulation. The authors hence concluded a network lifetime improvement of 12%. Other results regarding e.g. network performance were however not presented by the authors.

Nurmio et al. followed a similar idea of combining a link metric with an energy metric in their work "Equalizing Energy Distribution in Sensor Nodes through Optimization of RPL"[27]. Instead of the weighted approach from Chang et al. their proposed OF considered ETX as a first priority and the parent remaining energy of the node as the second. Instead of MRHOF-ETX, Nurmio et al. however compared their OF, of which there are two variants PEOF and PEOF2, with MRHOF-Energy. Their conclusion was once again an improvement in network lifetime. Despite that in a network where nodes are further apart, MRHOF-Energy was still shown to be a viable option as it kept more parts of the network reachable for longer which was displayed by the comparison of PDR.

Iova et al. presented a new approach to save energy in "Using Multiparent Routing in RPL to Increase the Stability and the Lifetime of the Network"[28]. They identified two main approaches until now. One is to minimize global energy consumption by choosing energy-efficient links. This approach may, however, cause nodes with the best links to deplete their energy faster as they are chosen more frequently to route packets. The second approach is to have nodes with a large residual energy to forward most of the traffic. This will, however, cause these nodes to run out of energy faster.

Iova et al. therefore proprosed an Expected Lifetime (ELT) metric which presents the estimation of the expected lifetime of a node if traffic stays the same as at that moment. The proprosed OF would then, based on ELT, choose the path that keeps the weakest node alive the longest, thus maximizing the lifetime of constrained nodes, through which many other nodes route their traffic leading to congestion. Such nodes are called bottlenecks. Additionally to the use of ELT Iova et al. introduced a second variant of their OF which makes use of a multipath approach, splitting traffic probabilistically between different parents.

They tested the performance of their proprosal in a WSNet simulation using 50 nodes, including one root, in a random topology, spread over a 300 $\times$ 300 m_2 area. The OFs which were compared with each other were the custom energy OF by Kamgueu et al.[25], MRHOF-ETX, the ETX and energy combination OF by Chang et al.[26], the by Iova et al. proposed ELT based OF and the ELT based OF with the addition of the multipath mechanism. From here on, we will call them the Kamgueue OF, MRHOF-ETX, the Chang OF, the ELT OF and the Multi ELT OF from here on.

The experiment results showed the Kamgueu OF with a considerably low PDR compared to the other OFs as it focused solely on residual energy and not on link quality at all. MRHOF-ETX which only pays attention to link quality had the highest PDR values. The remaining three OFs performed relatively well although the Multi ELT OF performed a little better than the Chang OF and the ELT OF.

In terms of end-to-end delay all OFs produced acceptable time spans where none stood out negatively.

Routing stability was considered by the authors by comparing the number of preferred parent changes that occurred during the simulation. Only the results for ELT OF and Multi ELT OF were presented in the paper and Multi ELT OF managed a much lower amount of preferred parent changes. This was interpreted as Multi ELT OF being more stable.

The last compared metric was the impact on energy efficiency on each OF. For energy efficiency, no data was presented for the ELT OF, only the Multi ELT OF was compared to the other OFs.

The Kamgueu OF achieved the lowest energy consumption by far. However, when the low PDR of the Kamgueue OF is also taken into consideration the low energy consumption may be due to less packets being forwarded.

Aside from the Kamgueu OF, the Chang OF consumed less energy than MRHOF-ETX, as it also pays attention to remaining energy, not only link quality as MRHOF-ETX does. Multi ELT OF consumed the most energy overall, especially in control messages during the start of the experiment as well as due to the active multipath traffic forwarding. It, however, makes up for that by achieving a much longer network lifetime compared to all other OFs.

At present, energy focused OF research has diversified into application-specific composite designs, demonstrated by Sun et al. in "Collaborative WSN-UAV Data Collection in Smart Agriculture: A Bi-objective Optimization scheme"[29]. The paper researched collaborative WASN-UAV data collection in smart agriculture with the same goal of maximizing the overall data transmission rate of the network while preserving the remaining battery energy of active sensor nodes. Although for a specific application, the core conflict remains finding a balance between network performance and energy efficiency.

3.1.4 Positioning of This Work

The reviewed literature includes simulation comparisons of standardized OFs OF0 and MRHOF. These comparisons were then transferred over to physical testbed experiments with actual sensor boards. These are, however, few and far in between and simulation experiments still dominate OF researches.

We therefore plan to invest our time in physical hardware experiments, both on a remote testbed as well as experiments done here at our institution. Most of the simulations are also conducted through the Cooja [17] simulator, a built-in network simulator for Contiki OS [16]. We will implement our OFs as well as our experiments on RIOT OS [3].

As improvements in network performance were achieved, often through link quality, nodes used more energy to sustain these demands in performance. Hence, many studies then moved their focus to improving the energy efficiency of sensor networks.

Our goal in this thesis is to implement MRHOF with the metrics ETX and LQI, of which LQI has been scarcely researched as a routing metric, as well as the energy. While many studies have focused on energy aware routing and even compared their proprosed energy aware OFs to MRHOF-Energy, a standardized MRHOF-Energy itself is seldomly the primary object of investigation and more of a secondary baseline.

3.2 Problem Analysis

Currently, the RPL implementation of RIOT OS [3] on the master branch, containing the current and official version of RIOT OS [3], only includes OF0.

3.2.1 Objective Function 0 (OF0)

OF0 [10] is the first OF standardized by the IETF and is entry 0 in the Objective Code Point (OCP). It bases its rank on the routing metric Hop Count (HC) which counts the number of routers a packet has to pass in order to traverse from its sender node to its destination. In OF0 the HC measured and used for rank calculation is the HC between the root and each node. OF0 aims for a node to join a DODAG version that offers good enough connectivity to a specific set of nodes or a larger routing infrastructure. However, there is no guarantee that the path will be optimized according to a specific metric.

Rank Computation The Rank is calculated by this formula:

$R(N) = R(P) + rank\_increase$,

where

- R(N): the node rank
- R(P): the rank of its selected preferred parent
- Rank increase: $rank\_increase = (R_f \cdot S_p + S_r) \cdot MinHopRankIncrease$

where

- S_p / step of rank: an intermediate value representing link properties with a neighbor, in this case HC
- R_f / rank factor: a multiplier used to scale link properties and reflect preference between link categories

- S_r / stretch of rank: a configurable parameter to exaggerate its own rank
- MinHopRankIncrease: a scaling factor configured by root which defines the absolute minimum increase in rank allowed

Parent Selection The node will calculate its rank based on each prospective parent and then choose the parent which offers the lowest rank as their preferred parent. The node will then route their upward traffic through the preferred parent. The node will also choose a backup feasible successor which they will route their upward traffic to if connection to the preferred parent fails.

As OF0 delivers the shortest path, it profits from its short computing time and fast network convergence because it relies on HC rather than dynamic link measurements. However, no attention is paid to link quality or energy depletion. OF0 therefore is an efficient option with simple implementation, but pays no attention to any other optimization goals.[10]

In order to expand the RIOT OS [3] RPL module and enable routing decisions based on other routing metrics, we have decided to add another OF to the RIOT OS [3] RPL module.

3.2.2 Minimum Rank with Hysteresis Objective Function (MRHOF)

MRHOF [11] is the second standardized OF of the RPL protocol and bears value 1 in the OCP. Its primary goal is to select routes that minimize a chosen routing metric while maintaining network stability.

Any MRHOF implementation will have to decide which routing metrics to implement for their MRHOF implementation. More than one routing metric can be implemented in MRHOF or added to the existing ones. Upon using MRHOF as an OF, any of the routing metrics offered by the implementation can be chosen. It is, however, only possible to use one routing metric at a time.

Consideration of Routing Metrics

The metric used for rank calculation by MRHOF may be chosen and exchanged as long as this metric can fulfill the MRHOF objective to minimize the given routing metric and is listed in RFC6551 [30]:

- ETX: This is the default metric used by MRHOF. It is the number of transmissions a node expects to make to a destination to successfully deliver a packet and receive an acknowledgement.
- HC which is also the default metric for OF0. It is the number of traversed nodes along a path.
- Latency: which represents the cumulative delay of the path in microseconds.

Parent Selection For each of its neighbors, which are prospective parents, a node will calculate the total path cost to the root by summing up the selected routing metric value to its parent and its parent path cost. In other words, the path cost is a sum of the selected metric values of all nodes passed on the way from the root node to this node.

The node will select the neighbor with the lowest path cost as its preferred parent. It thus seeks to minimize the chosen routing metric. In order to maintain network stability, it will only switch to a new preferred parent if the new path cost is lower than the current preferred parent path cost by a set threshold, a variable called PARENT_SWITCH_-THRESHOLD. This mechanism is called a hysteresis.

A node can also choose up to $PARENT\_SET\_SIZE - 1$ additional parents beside its preferred parents. One of the slots in its parent set is already occupied by the preferred parent. The node could route to its other parents when link or transmission failure or traffic congestion occur.

Accordingly, different from OF0, MRHOF bases its routing decision on a routing metric instead of mere distance. The function of the hysteresis mechanism is to prevent excessive churn on the network by hindering a node from changing their parent too often in a short amount of time.

Rank Computation After a node has evaluated all its prospective parents by calculating their path cost and decided on a preferred parent and added parents to add to its parent set, it sets their rank to the maxmimum of three values:

1. The rank calculated for the path through the preferred parent. The rank is calculated based on the path cost.
2. The rank of the parent with the highest rank: Adding this condition causes the node rank to be greather than the Rank of any parent in the parent set. If a node had to route a package to a parent in its parent set that is not the preferred parent, while the node has a higher rank than that parent, the node risks creating a routing loop. Therefore, the node rank should be higher than the rank of all parents in the parent set.
3. Largest calculated Rank among paths through the parent set, minus MaxRankIncrease: This caps the rank to not exceed the maximum allowed increase relative to the parents of the node.

3.2.3 Non-standardized Objective Functions

In addition to OF0 and MRHOF, which were standardized by the IETF, there are several other OFs [20] [31] which have been researched and tested.

Both OF0 and MRHOF are single metric OFs which make use of the value of exactly one metric to determine the node rank and select its parents. There are other single metric OFs, which make use of the aforementioned metrics in other ways or focus on other metrics.

Besides single metric OFs, there are multiple metric OFs which use different methods to determine the node rank and choose its parents. These methods include but are not limited to the following methods:

- Fuzzy logic method: Combine multiple types of input data into a single path quality score. The path quality score is used by the node to compare its prospective parents.
- Additive composite method: Sum of weighted metric values to turn multiple metrics into a single value.

- Lexical composite method: Evaluating different metric values sequentially in a hierarchical order.
- Context-aware approach: Routes data packets based on the surrounding situation using, e.g., weighted aggregation, fuzzy logic, machine learning or rule-based reasoning.

3.2.4 Approach to Solution

RIOT OS [3] currently only offers OF0 as an OF in the RIOT OS [3] RPL module. However, OF0 is implemented to only use HC for rank calculation and parent selection. No other routing metrics such as link quality are taken into consideration. If a node were to discover a parent with a low HC to the root, it would choose this node as their parent regardless of poor link quality or unreliable packet transmissions.

MRHOF may solve this problem by choosing routes based on a link quality aware metric. By choosing a different routing metric for MRHOF to work with, different constraints can be taken into consideration one at a time. Due to these reasons, we have decided to implement MRHOF to be added to the RIOT OS [3] RPL module. In order to have our MRHOF implementation integrate link quality consideration, these two routing metrics are part of the implementation and can be chosen from:

- ETX: Predicts the total number of packet transmissions, including retransmissions, required to successfully deliver a packet to its destination.
- LQI: Assesses the reliability and strength of a received radio signal [32]. LQI ranges from values 0 to 255, with 255 being the best and 0 being the worst value.

The node ETX value and LQI value to its parent can be found in the *netstats_nb_t* struct of the netstats neighbor module in the RIOT OS [3] implementation.

As throughput is listed in RFC6551 [30], we were previously also considering throughput to implement as a metric that can be chosen for MRHOF. Throughput measures the rate of succesful data transmissions in byte per seconds.

ETX and ETX values can be measured using control messages, which are sent periodically when nodes use RPL. However, throughput can only be observed when data is actively being sent. In LLNs links are often idle, meaning that data isn't frequently transmitted between two nodes at any time. It is possible for a node to go to sleep if it has no data traffic to forward and no data of its own to send. On such a link it will be difficult to measure and record meaningful throughput values.

If we wanted to force throughput measurements, we would have to make nodes generate extra data to send purely for the purpose of throughput measurements. This directly contradicts the challenges of LLNs which we are trying to solve. We want to minimize energy consumption and keep the routing path as unoccupied as possible so other nodes can use it. For this reason, throughput was considered impractical to implement as a routing metric.

While our existing MRHOF implementation now offers two metrics, neither ETX nor LQI considers the aspect of energy consumption which is a scarce resource that attention needs to be paid to in LLNs. Therefore we have decided to concentrate our efforts on implementing an energy based metric in our MRHOF implementation as well.

There are three types of energy sources [30] to differentiate:

- Mains-powered: Nodes connected to a constant power source are mains-powered.
- Battery-powered: Nodes that have a battery attached are battery-powered.
- Energy scavengers: Nodes that capture and convert ambient energy from the surrounding environment, such as from vibrations or light, are called energy scavengers. The amount of energy drawn from energy scavengers is often fluctuating and not reliably available at any time.

The energy formula we used and specifications of our energy sources will be further explained in Section 4.2.

4 Implementation

4.1 Implementation of MRHOF

There have been previous attempts made to add MRHOF to the RIOT OS [3] main repository. Therefore Pull Request (PR) 14623 [33] from July 2020 was used to base our implementation on. As the RIOT OS [3] implementation has evolved further in the past years, we first had to revise the last draft and adapt it to the latest RIOT OS [3] version.

Fig. 4.1 shows the function call diagram of the implementation.

The module is accessed by calling the function *gnrc_rpl_get_of_t* which returns a pointer to a static object *gnrc_rpl_get_of_mrhof*. *gnrc_rpl_get_of_mrhof* is of type *gnrc_rpl_of_t*. This OF representation in RIOT OS [3] holds function pointers to *calc_rank*, *which_parent*, *which_dodag*, and *reset*.

These five functions are the entry points of the MRHOF module as they are called by RPL functions outside of the MRHOF module.

Figure 4.1: Function call diagram of MRHOF functions

`uint16_t calc_rank(gnrc_rpl_dodag_t *dodag, uint16_t base_rank)` calculates the node rank based on the rank of the parent and the chosen metric. However, the parent this function is used for will only be considered if the calculated path cost is lower than the path cost of the current parent by at least PARENT_SWITCH_THRESHOLD. This is the aforementioned hysteresis mechanism and protects the node from switching its parent too often.

`int which_parent(gnrc_rpl_parent_t *p1, gnrc_rpl_parent_t *p2)` compares a first parent p1 and a second parent p2 and returns 0 if the parents are of equal preference, a positive value if the second parent is chosen over the first and a negative value if the first parent is chosen over the second. The parents in the parent set of a node are ordered according to these comparisons. The parent with the highest placement in this ordering is the preferred parent. The function compares two parents based on the following criteria:

1. Metric value acceptability

   A parent is acceptable if its metric value is not higher than the defined parameter MAX_LINK_METRIC.

2. Freshness

   The netstats neighbor module contains a function to check the freshness of the values. This value is used to determine how recent the recorded data is. If a significant amount of time has passed without any communication from or to a node, the values recorded are deemed not fit for usage.

3. Rank

   Lastly, parents are compared based on the chosen metric value and the resulting path cost from each parent. In *which_parent*, p1 is always the higher ranked parent at the moment and p2 may only be deemed a better parent than p1 if the condition of the hysteresis is fulfilled.

`int which_dodag(gnrc_rpl_dodag_t *d1, gnrc_rpl_dio_t *dio, kernel_pid_t dio_iface, ipv6_addr_t dio_addr)` compares two DODAGs, the current DODAG of the node and a new DODAG from which they got a DIO message. The function returns a positive value if the new DODAG is preferred by the node. The node will then leave its old DODAG and switch to this new DODAG. If the function returns a negative value, the current DODAG is preferred by the node and the node will stay in this DODAG. If 0 is returned, the node views the two DODAGs as equally preferrable and will remain with its current DODAG. The following are the criteria for the comparison:

1. grounded: whether a DODAG is grounded
2. Preferable root:

   Each DODAG root within a RPL instance is assigned a DODAG preference in the range of 0 to 7, 0 meaning the least preferred and 7 the most preferred. The preference score is influenced by whether the DODAG is grounded or floating, administrative preferences, the OF and routing metrics. In general, nodes prefer a more preferable DODAG root.
3. rank:

   The two DODAGs are compared based on the rank they would obtain within each DODAG. The node will only switch to the new DODAG if the calculated path cost fulfills the hysteresis condition.

Lastly, `void reset(void)` is called to reset the internal state of an OF. As MRHOF keeps no persistent values that need to be cleared, its `reset()` implementation returns without performing any action.

Besides the five functions anchored as function pointers in $gnrc\_rpl\_get\_of\_mrhof$, `uint16_t _link_metric(netstats_nb_t *stats)` is responsible for retrieving the value of the selected metric used for MRHOF.

For each implemented metric, the function computes the routing metric, based on which routing metric was chosen to use in MRHOF.

- If the chosen metric is ETX, the function returns the ETX value from the RIOT OS [3] netstats neighbor module.
- If the chosen metric is LQI, the function retrieves the current LQI value from the RIOT OS [3] netstats neighbor module. This LQI value is then converted to an ETX value from 1 to 7, with 1 being the best and 7 being the worst value, using this formula:

  $3\dot{(}0xFF - LQI_P) + NETSTATS\_NB\_ETX\_DIVISOR$

where LQI_P is the LQI value of the link to the parent and NETSTATS_NB_-ETX_DIVISOR is the divisor constant 128. ETX values are stored as $ETX_{realvalue} \cdot 128$ [30]. In order to get a real ETX decimal value, it first needs to be divided by the NETSTATS_NB_ETX_DIVISOR.

So NETSTATS_NB_ETX_DIVISOR represents exactly one ETX unit. It is added here to represent the minimum ETX of one. The rest of the formula scales the LQI value into the same range of $ETX_{realvalue} \cdot 128$.

255 represents the best LQI value while 0 presents the worst. An ETX value of 1 represents the best ETX value with a good link quality, while high ETX values represent a bad link quality. This enables a linear mapping, as good link quality is indicated by a low ETX value and a high LQI value and a bad link quality is indicated by a high ETX value and a low LQI value. So the LQI value is deducted from 255 in order to invert the LQI value.

Fig. 4.2 displays this linear conversion.

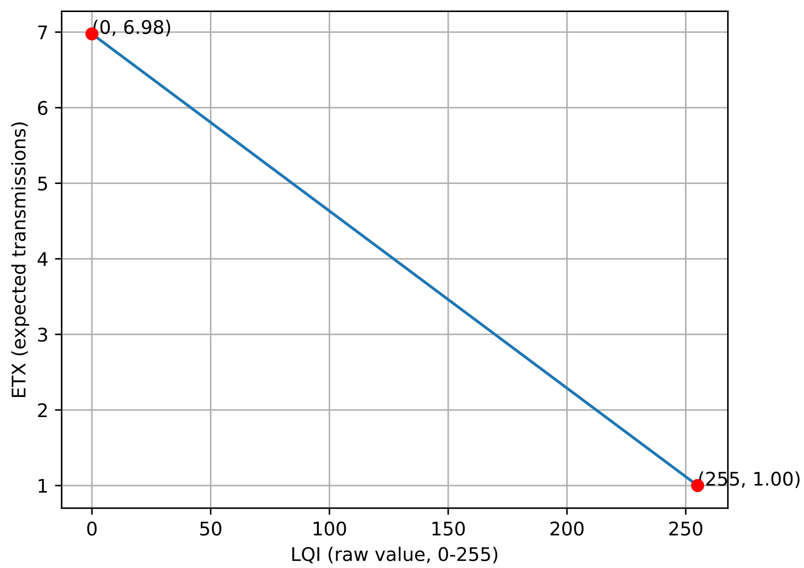


Figure 4.2: Linear function which converts LQI values to ETX values

- If the chosen metric is energy, the function retrieves the value of a metric called energetic happiness from the function *get_energetic_happiness*. A node with an energetic happiness value of 100 is considered good and should be chosen as a parent, while an energetic happiness value of 0 is considered bad and a node with this energetic happiness value should not be chosen as a parent. Further explanation to this metric is provided in Section 4.2.

This value is then converted to a ETX value from 1 to 7, with 1 being the best and 7 being the worst value. The formula for this linear function is

$$3 * (100 - E_E) * 255/100 + NETSTATS\_NB\_ETX\_DIVISOR$$

where the function E_E is the return value $get\_energetic\_happiness$. The mapping to ETX follows the same pattern as the one of LQI.

Fig. 4.3 displays this linear conversion.

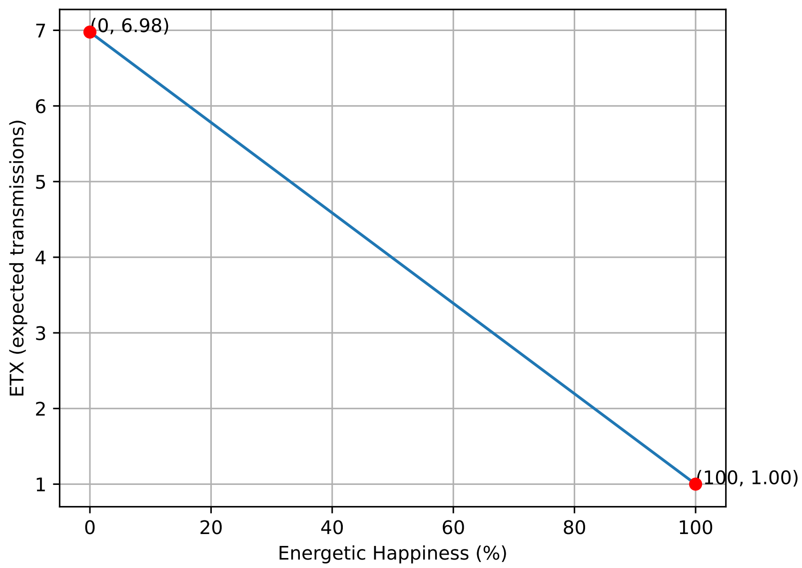


Figure 4.3: Linear function which converts energetic happiness values to ETX values

The conversion of other metric values to ETX values was implemented because MRHOF uses the following parameters [11] which are necessary for rank calculation and parent selection. These values are based on the IETF Routing over Low power and Lossy networks (ROLL) working group experiences and are recommended in RFC 6719 [11] when ETX is the selected metric.

- **MAX_LINK_METRIC**: defines the maximum allowed value for the selected link metric for each link on the path and is set to 512.

- **MAX_PATH_COST**: defines the maximum allowed value for the path metric of a selected path and is set to 32768.
- **PARENT_SWITCH_THRESHOLD**: defines the difference between the cost of the path through the preferred parent and the minimum cost path in order to trigger the selection of a new preferred parent and is set to 192.
- **PARENT_SET_SIZE**: defines the number of candidate parents, including the preferred parent in the parent set and is set to 3.

4.2 Implementation of Energy Metric in MRHOF

Following section 3.2 of RFC6551 [30], which provides further information about routing metrics, energy was added as an additional routing metric to MRHOF to choose from.

Provided are one formula for energy scavengers and two for battery-powered nodes. As our available testing board, the Adafruit feather nRF52840 sense, has a battery terminal built in, we focused on implementing one of the formulas for battery-powered nodes. The first formula required the node to be able to measure its average power consumption. The second formula required the node to be able to measure the current energy in the battery and the total elapsed time.

We decided to implement the second formula as we considered the requirements easier to fulfill. The remaining energy in the battery could be measured using the RIOT OS [3] Peripheral Driver Interface module to access the built in Analog-to-Digital Converter (ADC). The elapsed time could be measured using the RIOT OS [3] ztimer module.

$$E_E = T \cdot \frac{E_{bat}}{E_0 \cdot (T - t)} \tag{4.1}$$

where E_E is called the Energetic Happiness of the node and represents the current expected lifetime divided by the desired minimum lifetime in units of percent. E_{bat} is the currently remaining power, E_0 the inital power at start of runtime, t the total elapsed time and T the desired lifetime.

The implementation of the energy metric using a battery as its power source therefore has the following functions

- `uint32_t get_remaining_energy(void);`
  This function measures the current energy of the battery.

- `uint8_t get_energetic_happiness(void);`
  This function calculates the current energetic happiness of the battery.

- `void init_mrhof_energy(void);`
  This function is called at the start of runtime to capture the initial time and the initial energy of the battery for later calculations.

and uses the following parameters

- **DESIRED_LIFETIME_US** is the desired lifetime and has to be adjusted according to the battery and the network requirements.
- **uint32_t initial_energy** and **uint32_t initial_time** have to be captured at the beginning of runtime and are used to calculate the energetic happiness.
- **ADC_RES, ADC_MAX_VAL, ADC_VREF_MV, ADC_BAT_RESISTOR_DIV, AVG_CNT, VBAT_MIN_MV, VBAT_MULTIPLICATOR** and **ADC_BATTERY_LINE** are used to estimate the battery remaining energy by its current voltage.

Battery In order to implement and later assess MRHOF using energy as the routing metric, some battery testing had to be conducted. The batteries used were lithium polymer batteries with a voltage of 3.8 V, a capacity of 600 mAh and a discharge rate of 36 C. Using the RIOT OS [3] Peripheral Driver Interface, an estimated voltage value could be measured.

As no data sheet was available for the batteries, we had to determine a discharge curve ourselves through manual testing in order to determine the battery runtime and set appropriate values for the parameters used for voltage measurement and the calculation of energetic happiness.

One issue was the board starting the charging process of the batteries as soon as the board was connected to a computer using the USB-UART port, which is why data had to be transmitted wirelessly from node to node to receive measurement values during testing as well as the experiment. Fig. 4.4 displays the battery discharge curve while the board was merely sending messages using the RPL protocol. So the battery may discharge faster than the curve indicates when in actual use.

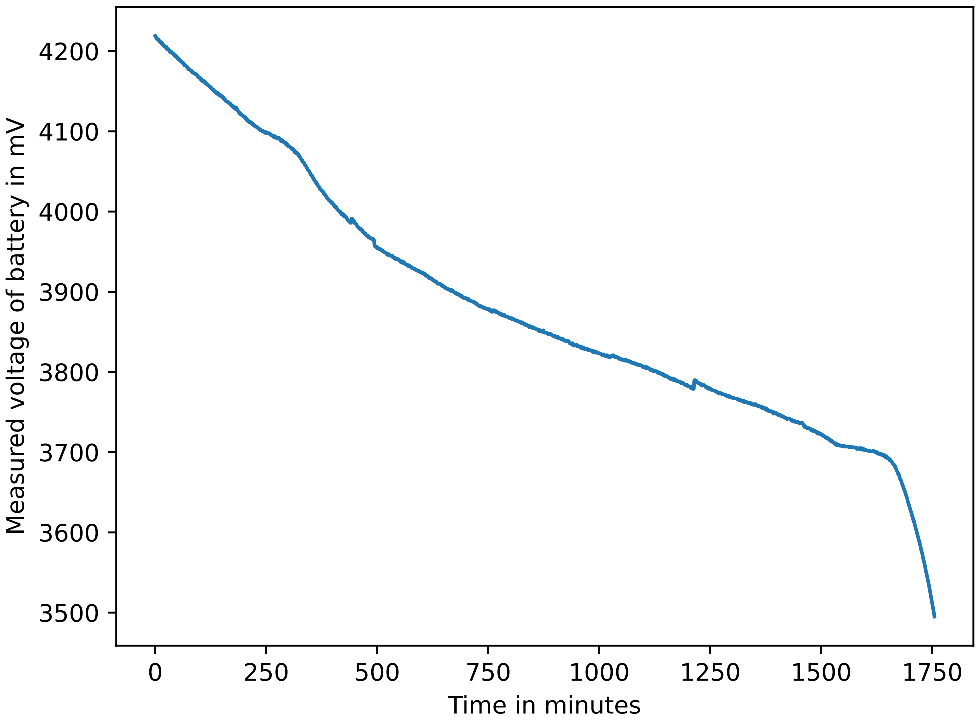


Figure 4.4: Discharge curve of battery

5 Evaluation

5.1 Experiment Execution

In order to compare different OFs, we implemented a benchmark application based on the UDP Benchmark module [34] [35] available in RIOT OS [3].

Our experiment includes one root node and at least one non-root node. The non-root nodes periodically send ping messages to the root.The root then answers with a response pong message. When the non-root nodes receive the pong response from the root node, they calculate the RTT of the message exchange and send the measurement as well as further information with their next ping message to the root. All this data is collected at the root and saved in a log file. The log file is then later analyzed using a script. In order to build up the DODAG, non-root nodes first wait until they have joined a DODAG before they start sending ping messages. Fig. 5.1 illustrates the sequence of the experiment.

As nodes further away from the root may be too far away from the root to directly reach the root, they cannot choose the root as their parent and have to wait for other nodes closer to the root to join the DODAG in order for them to choose these nodes as their parents and join the DODAG. So non-root nodes may start sending ping messages at different times as they may join the DODAG at different times.

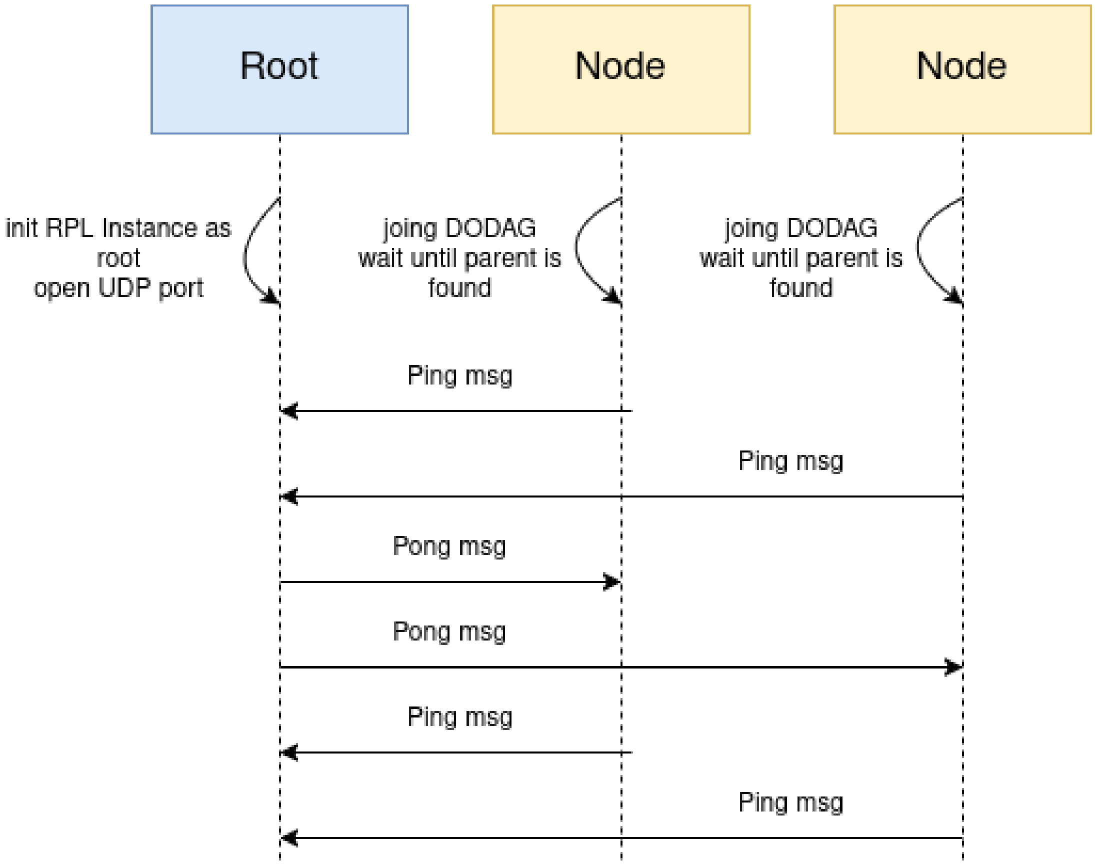


Figure 5.1: UML sequence diagram of message exchange between root and other nodes during experiment

Witch each ping, the non-root nodes provide the root node with the following information:

- Their own local time: Each node starts a timer when they start sending messages and provide the root with the time that has elapsed since then for later analysis. As each node has its own clock, time synchronization across every node would require a lot of effort. So we decided to have each node send their own local time at the time they send a message.

  If the time of the root node was used as a global time, it would produce imprecise data as well, as the root cannot process more than one message at a time. Messages from many nodes may be waiting in a queue for processing and a delay of waiting time would be added on top.

- RTT: RTT is calculated by subtracting the time the ping message was sent from the time the pong message arrived.
- ETX and LQI: Routing metric measurements are provided by each node via the RIOT OS [3] netstats neighbor module.
- HC
- the number of ping messages they have sent to the root
- the number of pong messages they have received back
- their current parent
- their current battery level if available

The root node receives this data in a thread which is tasked only with receiving ping messages and sending pong messages back immediately after. The message content itself is handed over to a different thread which is responsible for processing the message content. The root node was implemented in this way in order to separate the message content processing from the receiving and sending. The processing requires more time and the task of receiving and sending may be delayed if handled by the same thread. Fig. 5.2 showcases this separation of tasks.

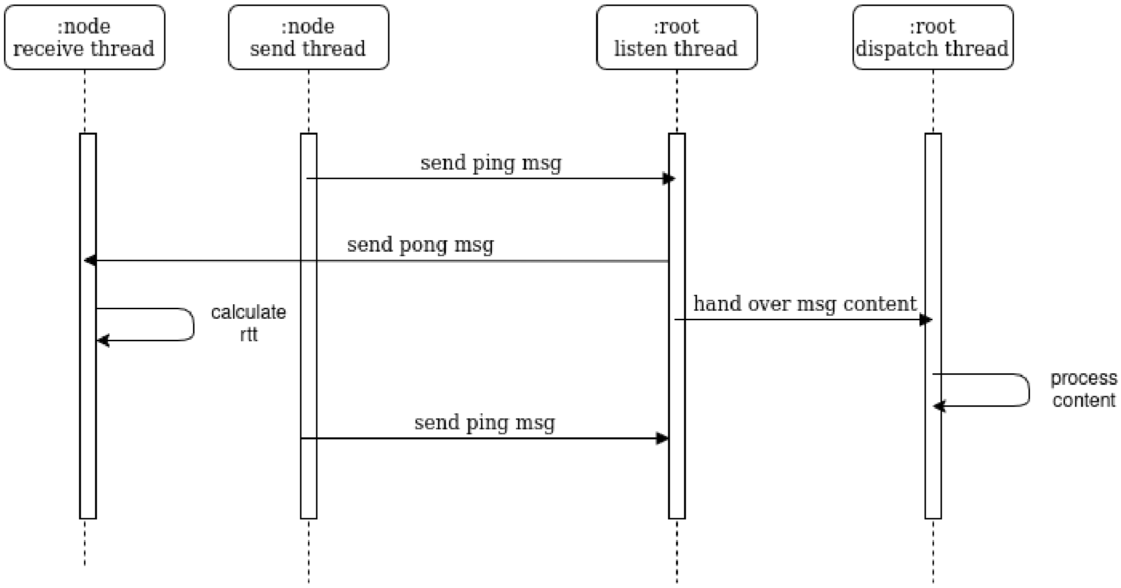


Figure 5.2: UML sequence diagram of message exchange between root threads and node threads during experiment

After the completion of the experiment, the data provided was extracted using a Python program which scanned the output from the experiment and summarized and aggregated the provided data. A Python program was also used to parse the output of children and parent pairs in the DODAG and rebuild the DODAG topology later on. The experiment results will be used to compare the different OFs according to performance metrics, such as PDR and RTT, and the built topologies.

An experiment run setup includes:

1. Deciding the OF used for this experiment run and adjusting test parameters in the applications.
2. Flashing and starting first the root node and then the other nodes.

Parameters In order to test the OFs under different conditions, a few parameters can be adjusted.

The number of nodes in the experiment can be changed to see how few or many nodes impact a DODAG. This information is also needed for the root node to reserve enough space to save information about the other nodes as well as for all nodes to reserve enough space to save information about its neighbors.

Another test parameter is the load the non-root nodes would send to the root. We tested the network with a load of a one-second, 500 ms and 50 ms interval between ping messages.

The number of pings each node was sending can be adjusted, shortening or prolonging the experiment duration.

All nodes sent packets exactly the size needed to contain all information in a ping which was 36 bytes. The 6LowPAN header added 41 bytes onto the packet and the Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 frame added 16 bytes on top. The whole packet to be sent therefore consisted of exactly 93 bytes which is below 127 bytes, the maximum physical packet size of the IEEE 802.15.4 standard which we were using. This enabled our ping message to be sent in one frame.

5.1.1 FIT IoT-LAB

Most of the experiments were conducted on the FIT IoT-LAB [1], a remote IoT testbed with several facilites throughout France. These testbeds provide fast remote access to different hardware boards.

Unfortunately, there were only two types of boards installed in the testbed whose drivers supported netstats neighbor statistics for ETX. Those were the nrf52840dk nodes and the samr21 nodes. Out of those two, only the nrf52840dk nodes, which are located in Saclay, were accessible at the time the experiment was conducted. Their small number of ten nodes in total only made a few different tests on this small amount of nodes possible.

Fig. 5.3 displays the placement of nodes at the Saclay testbed site used for the experiment.

The details of the experiment parameters are described in Table 5.1.

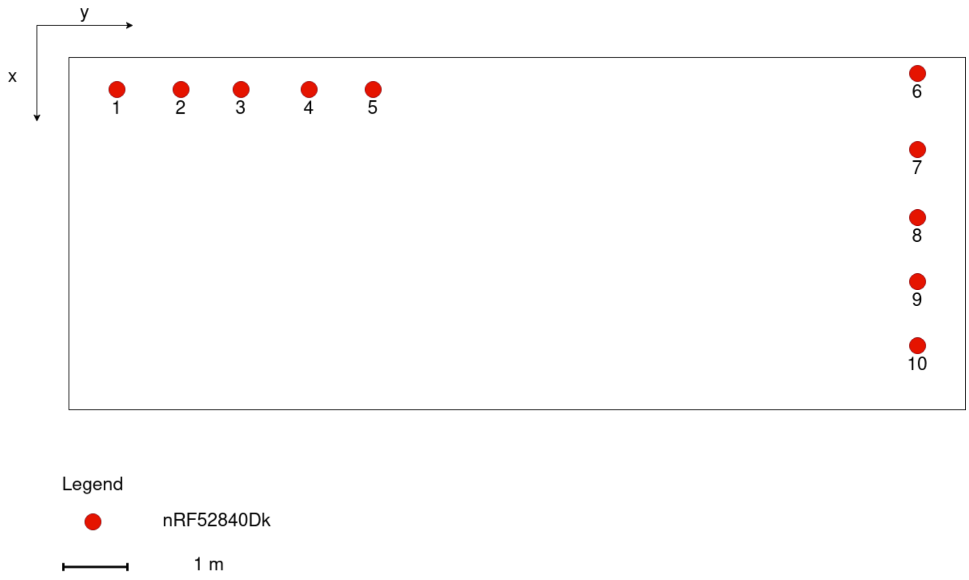


Figure 5.3: Placement of nodes at Saclay testbed site

Table 5.1: Experiment parameters for FIT Iot-LAB [1] experiment

Parameter	Value
Testbed Site	Saclay
Operating System	RIOT OS [3]
Number of nodes	10
Number of root nodes	1
Boards	nRF52840DK [13]
Experiment area	$5 \times 13\ m_2$
Traffic type	Constant Bit Rate (CBR)
Traffic rate	1 packet/min, 1 packet / 500 ms, 1 packet / 50 ms
Data packet size	93 bytes (including headers)
MAC Layer	802.15.4
Network layer	IPv6 + 6LoWPAN
Transport layer	UDP
Transmission Power	0 dBm
RPL Mode	storing
Traffic patterns	MP2P, P2P

5.1.2 Experiments with MRHOF-Energy

The only experiments that could not be conducted on the FIT IoT-LAB [1] were those employing MRHOF-Energy. As the nodes on the FIT IoT-LAB [1] lack internal battery measurements that are accessible to the firmware, no measurement of an energy status could be made.

There is the special feature of power consumption monitoring provided on the testbed. Its measurement values can be put into the formula which required the measurement of average power consumption to calculate Energetic Happiness. The testbed utilizes INA226 hardware components integrated into an additional control node attached to the actual experiment node. Due to this, measurements include the energy consumption of the MCU internal peripherals, system clocks and oscillators, board-level-voltage regulators as well as on-board LEDs. Consequently, this produces a higher energy consumption than one would normally expect from a program only sending and receiving packets wirelessly. Furthermore, this monitoring data has to be accessed through a separate monitoring profile and cannot be accessed by the nodes during runtime. As such, it could not be used to provide data to use energy as a routing metric for MRHOF-Energy.

We have therefore decided to implement MRHOF-Energy by attaching a battery to the Adafruit Feather nRF52840 Sense boards and conducting this experiment at the University of Applied Sciences Hamburg. For this purpose, the root node had to be attached to a computer via UART connection to capture the output of the experiment. Other nodes had to be flashed beforehand and distributed throughout the experiment space. Then each node had to be started individually by connecting the energy source.

In order to achieve different battery status for each node, some nodes carried out other tasks besides sending and receiving UDP packets, e.g., turning on LEDs.

The experiments to test MRHOF-Energy were changed to only include an uplink, no downlink. For this experiment, OF0 was employed to be compared with MRHOF-Energy. The focus of the experiment was energy consumption and the changes of the topology over the course of the experiment. The details of the experiment parameters are described in Table 5.2.

Fig. 5.4 shows the placement of nodes for the experiment.

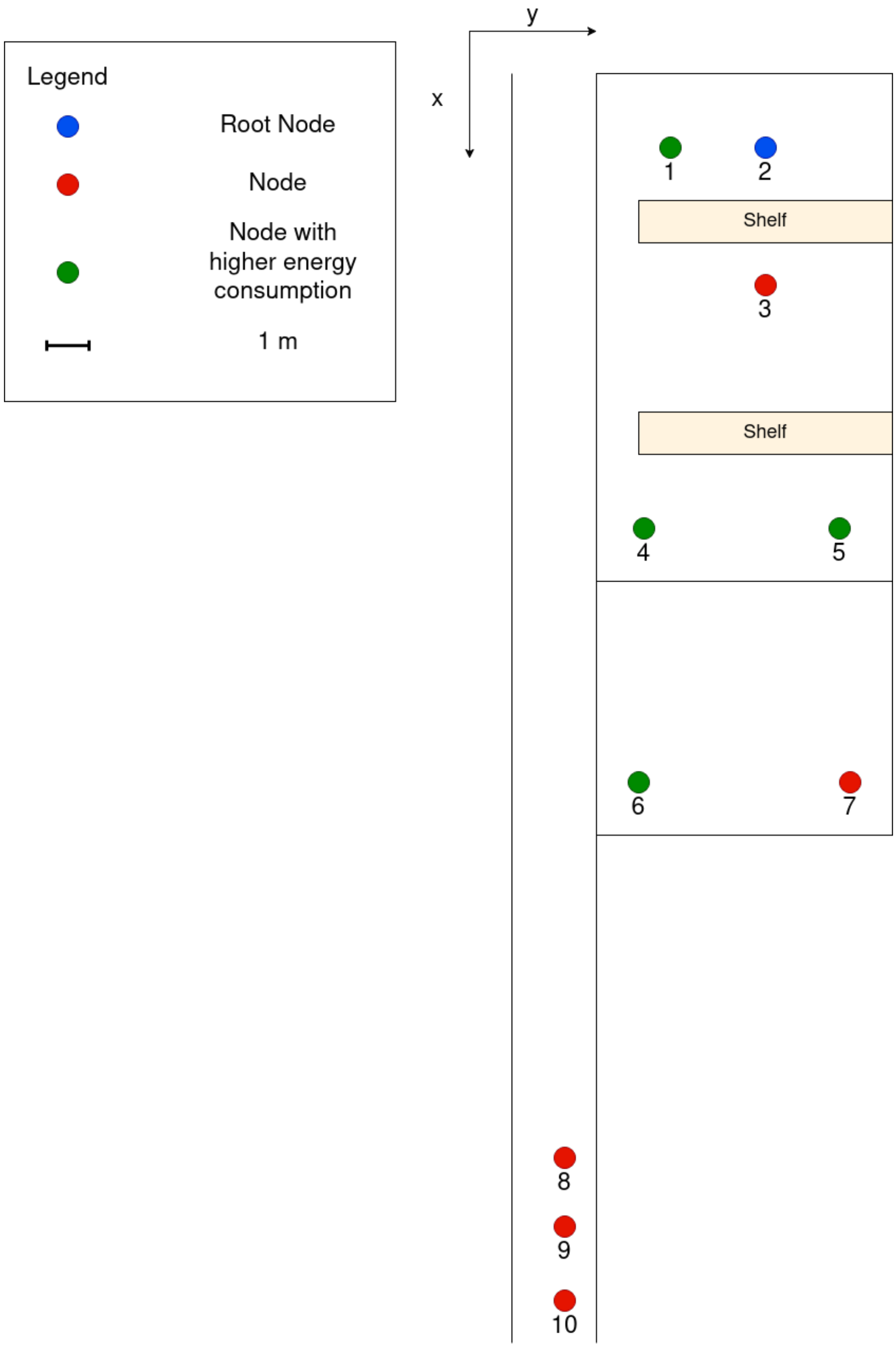


Figure 5.4: Placement of nodes during energy experiment

Table 5.2: Experiment parameters for experiment using batteries

Parameter	Value
Operating System	RIOT OS [3]
Number of nodes	10
Number of root nodes	1
Boards	Adafruit Feather nRF52840 Sense [12]
Experiment area	$8 \times 30\ m_2$
Experiment duration	4 hours, 2 minutes and 17 seconds
Traffic type, rate	CBR, 1 pkt/min
Data packet size	77 bytes (including headers)
MAC Layer	802.15.4
Network layer	IPv6 + 6LoWPAN
Transport layer	UDP
Transmission Power	0 dBm
RPL Mode	storing
Traffic patterns	MP2P

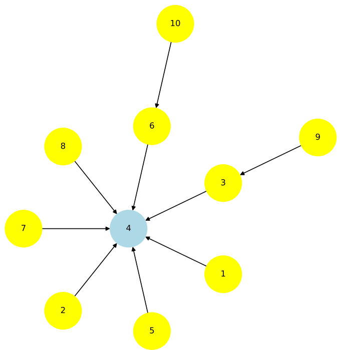
5.2 Results

This section will present the experiment results grouped by different network metrics, many of which were introduced in Table 2.1.

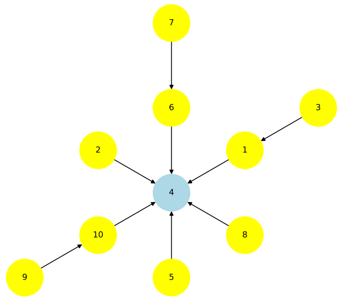
5.2.1 Network Topology

Although the network topology is not a metric itself, the network behaviour can be explained and conclusions to metrics can be drawn. Therefore, we will take a look at the different network topologies which resulted from different OFs.

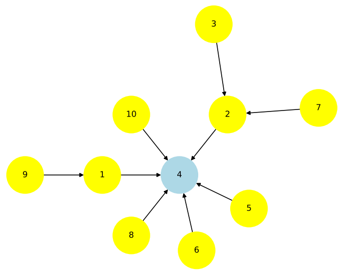
Fig. 5.5 shows the topology at the end of the experiment for a load of 1 packet per 1 s for each OF. Each node is marked with the same number from fig. 5.3. All topologies are in a similar pattern with the majority of nodes choosing the root node as their preferred parent and only a few multiple hop nodes as part of an outer circle.



(a) Network topology of OF0



(b) Network topology of MRHOF-ETX

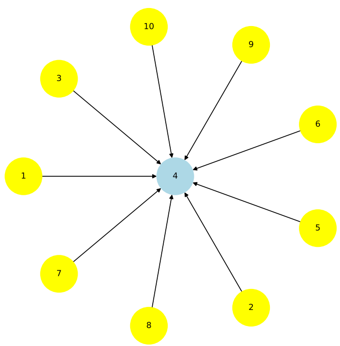


(c) Network topology of MRHOF-LQI

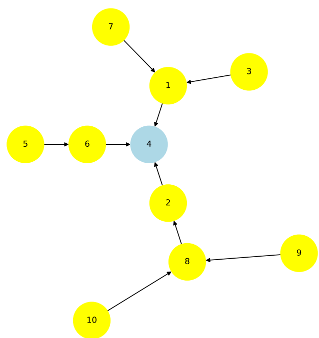
Figure 5.5: Network topology at end of experiment for 1 packet per 1 s load for OF0 (left), MRHOF-ETX (middle) and MRHOF-LQI (right). The root node is colored blue. All other nodes are colored yellow.

Fig. 5.6 shows the topology at the end of the experiment for a load of 1 packet per 500 ms for each OF. Each node is marked with the same number from fig. 5.3.

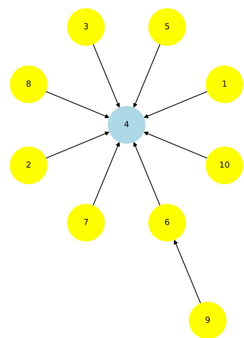
Under OF0, all nodes are arranged in a circle around the root node. MRHOF-LQI produces a similar shape with one two hop node. Meanwhile, MRHOF-ETX produced a more tree like shape with nodes that branch out and are more than one hop away from the root node.



(a) Network topology of OF0 for 1 packet per 500 ms load



(b) Network topology of MRHOF-ETX for 1 packet per 500 ms load



(c) Network topology of MRHOF-LQI for 1 packet per 500 ms load

Figure 5.6: Network topology at end of experiment for 1 packet per 500 ms load for OF0 (left), MRHOF-ETX (middle) and MRHOF-LQI (right). The root node is colored blue. All other nodes are colored yellow.

This observation is also supported by looking at the hop count of each node. Fig. 5.7 shows the average hop count for each OF.

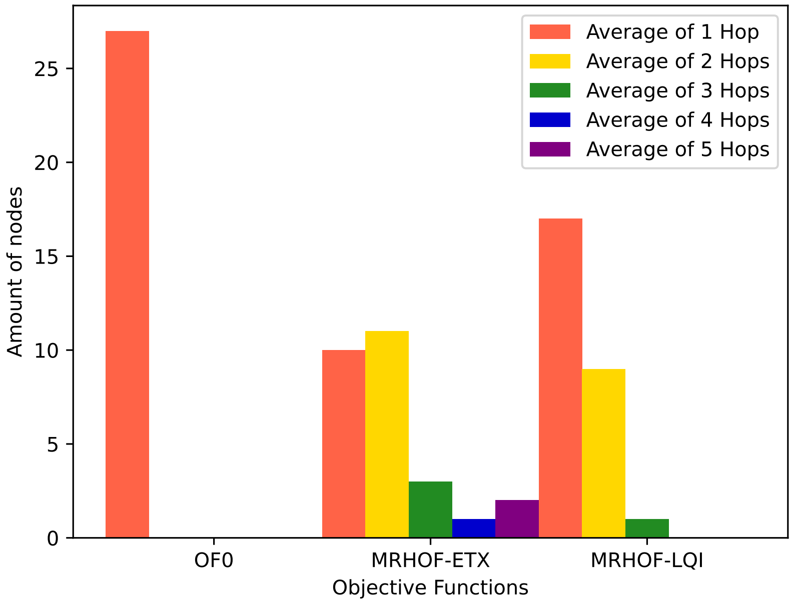


Figure 5.7: Amount of nodes which had average hop count indicated in the y-axis for OF0, MRHOF-ETX and MRHOF-LQI for all experiments on the Saclay testbed.

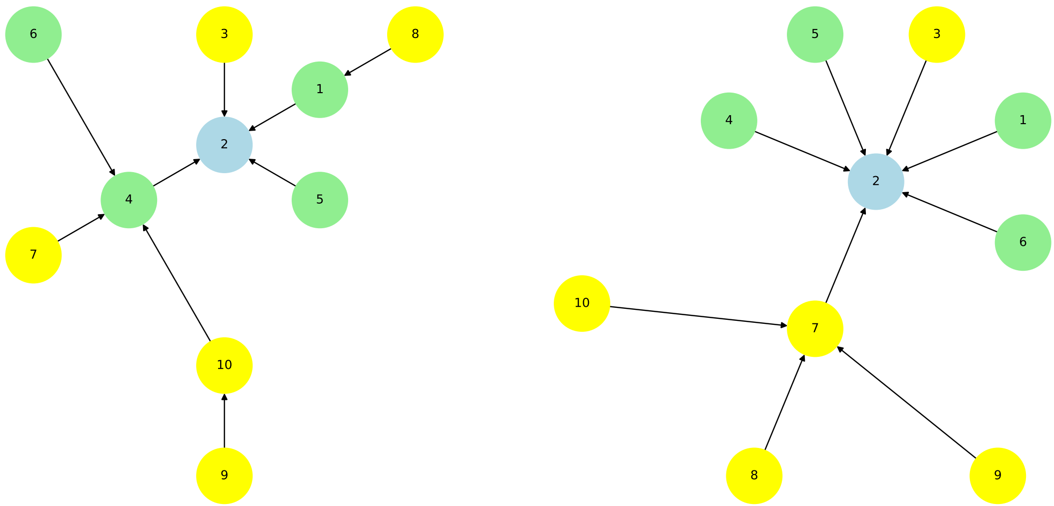
Under OF0, all nodes had an average hop count of 1. The nodes under MRHOF-ETX and MRHOF-LQI have more varied hop counts. Their packets actually travel more than one hop. Different from OF0 and MRHOF-LQI where the majority of nodes had a hop count of 1, under MRHOF-ETX, hop count 2 was the hop count most held by all nodes.

At a 50 ms packet interval, the OF0 topology is unchanged from Fig. 5.6a, while both MRHOF OFs assumed the shape shown Fig. 5.6c.

Fig. 5.8 shows the network topologies of OF0 and MRHOF-Energy at the end of the experiment conducted at the University of Applied Sciences Hamburg using batteries.

Although the same amount of nodes was used as in the experiment conducted on the Saclay testbed, the shape of the topology under OF0 is not in a circular shape, as is typical for OF0. However, this may be influenced by the longer duration and less frequent messages as well as the placement of nodes in this experiment.

Unlike OF0, under MRHOF-Energy, the nodes moved not to choose nodes which had consumed more energy during the experiment as their parent. Nodes farther away from the root node, such as node 8, 9 and 10, chose nodes with a lower energy consumption as their parent.



(a) Network topology of OF0 (b) Network topology of MRHOF-Energy

Figure 5.8: Network topology at end of experiment for OF0 (left), MRHOF-ETX (middle) and MRHOF-Energy (right). The root node is colored blue. Nodes which executed other tasks and consumed more energy are colored green. All other nodes are colored yellow.

5.2.2 Stability of Network

The number of times a node changes its preferred parent is indicative of the stability of a the network topology stability. A parent change occurs when a node decides that a different neighbor now offers a better path to the root. Frequent parent changes suggest that the network has not converged to a stable network topology. This can be caused by fluctuating link quality between nodes. A high rate of parent changes is generally not desired as each change is followed by control message exchanges and can cause temporary packet loss while a node changes from one parent to another. Hence, few parent changes indicate a stable topology, although that by itself does not guarantee a good performance.

Aside from the total number of parent changes, the distribution of parent changes across nodes indicates different causes. If a small number of nodes account for most of the changes, this indicates a local instability. The node may be struggling to find any neighbor offering a stable link. If parent changes occur across most nodes, its reason may be an OF with poorly chosen switching criteria.

Fig. 5.9 shows the recorded parent changes at the end of an experiment where each node sent 1000 messages to the root with a 1 second interval between each message. This data was chosen as an example as the number of parent changes according to the chosen load and duration of the experiment.

While few changes occurred across few nodes under OF0, more changes occured under MRHOF-ETX and MRHOF-LQI. There is, however, a difference between the two. Only a few nodes changed their parents with one node changing their parent very often under MRHOF-LQI. This points to a local instability or one or two nodes struggling to settle on a preferred parent. Under MRHOF-ETX the changes are spread more evenly across all nodes. Nodes are changing their parents quite often. Changing the hysteresis threshold for MRHOF may lessen the frequent parent changes.

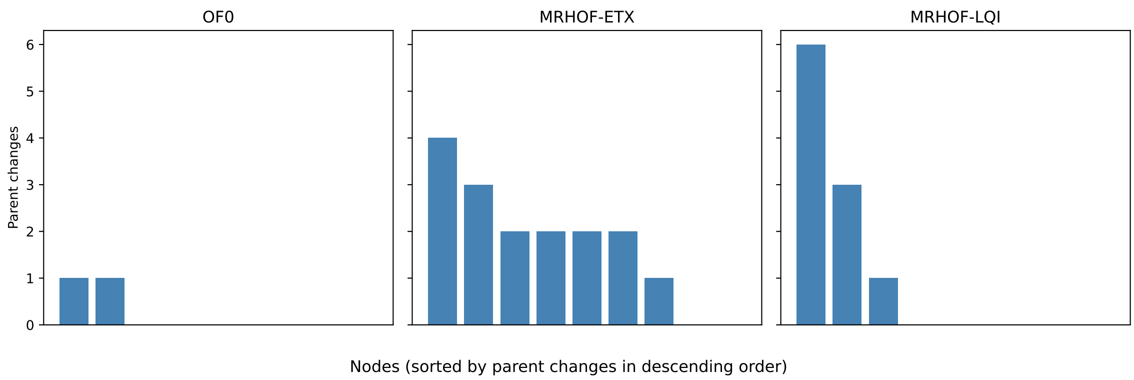


Figure 5.9: Parent changes of nodes sorted in descending order for OF0, MRHOF-ETX and MRHOF-LQI for a load of 1 packet every second for 1000 messages per node

These observations are also simmilarly reflected in Table. 5.3 which shows the average, maximum, minimum, standard deviation of parent changes as well as the percentage of nodes which had at least one parent change for each OF for the loads of packets being sent with a 1 second, 500 ms and 50 ms interval.

Table 5.3: Statistical data of parent changes for OF0, MRHOF-ETX and MRHOF-LQI

OF	Average	Maximum	Minimum	Standard deviation	% of nodes
1 s interval					
OF0	0.2	1	0	0.42	22
MRHOF-ETX	1.8	4	0	1.23	78
MRHOF-LQI	1.1	6	0	1.97	33
500 ms interval					
OF0	0	0	0	0	0
MRHOF-ETX	16.1	37	8	8.24	100
MRHOF-LQI	1.1	2	0	0.99	56
50 ms interval					
OF0	0	0	0	0	0
MRHOF-ETX	5.0	11	0	3.56	89
MRHOF-LQI	0	0	0	0	0
All intervals					
OF0	0.1	1	0	0.26	7
MRHOF-ETX	7.6	37	0	8.07	89
MRHOF-LQI	1.0	6	0	1.23	22

Fig. 5.10 shows the recorded parent changes at the end of the experiment which used batteries. MRHOF-Energy clearly shows a a more dynamic network topology than OF0 where nodes frequently change their parents. This is to be expected as the energy of nodes constanly changes over the course of the experiment while the HC of a node rarely changes.

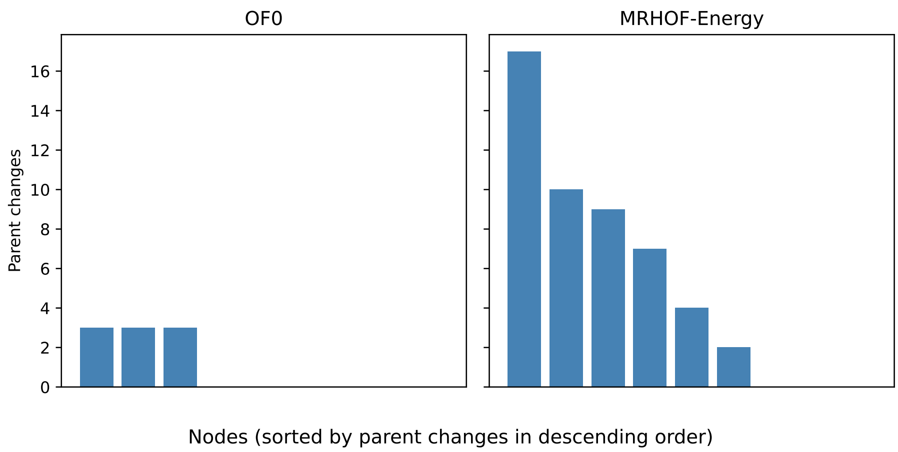


Figure 5.10: Parent changes of nodes sorted in descending order

This observation is also reflected in Table 5.4. The average number of parent changes of MRHOF-Energy indicates a higher instability within the network and nodes finding a more suitable parent over the course of the experiment.

Table 5.4: Statistical data of parent changes for OF0, MRHOF-ETX and MRHOF-Energy

OF	Average	Maximum	Minimum	Standard deviation	% of nodes
OF0	1.0	3	0	1.41	33
MRHOF-Energy	5.4	17	0	5.50	67

5.2.3 Packet Delivery Rate

The PDR metric is the rate of sent packets to successfully delivered packets. Link instability, the distance of nodes from each other and interfering signals from other devices cause frequent packet loss in LLNs. So guaranteeing a high PDR is an essential requirement of RPL network.

Fig. 5.11 shows the PDR of all messages within the network for all OFs for a load of a one second packet interval, a 500 ms interval and a 50 ms second interval. For a one second interval, all OFs achieve a similarly high PDR of almost 100%. For a packet interval of 500 ms, OF0 and MRHOF-LQI still achieve a PDR of 100%. Under MRHOF-ETX, the PDR drops to 70%. For a 50 ms interval, the PDR of OF0 and MRHOF-LQI is about 40% and that of MRHOF-ETX below 30%.

The gap in PDR from MRHOF-ETX to OF0 and MRHOF-LQI under high load can be attributed to the topology of MRHOF-ETX shown in fig. 5.6. In a small network as this one a circular topology can be more efficient for messages to reach the root node directly. In a tree-like topology, some routers have to work under stress, sending their own packets to the root node while routing packets from many other nodes at the same time. The router may not be able to route all packets reliably which leads to packet loss.

One more significant observation is the PDR of uplink and downlink respectively. Fig. 5.12 shows the PDR of these two routing directions for the packet loads of 1 s, 500 ms and 50 ms packet intervals for OF0, MRHOF-ETX and MRHOF-LQI. While the uplink mirrors the shape of the overall PDR in Fig. 5.11, the downlink PDR is almost consistently at about 100%. The only exception is the PDR of the downlink of MRHOF-LQI at a 500 ms packet interval. Otherwise, the downlink is unaffected by the changes in load.

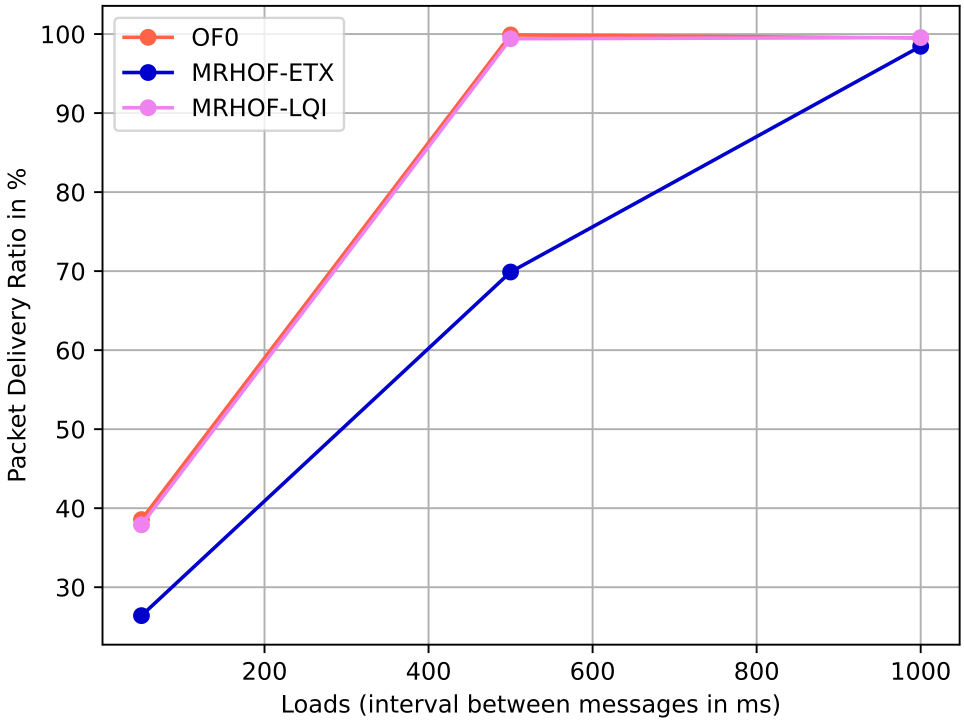
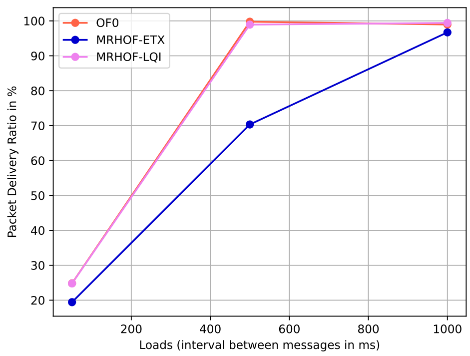
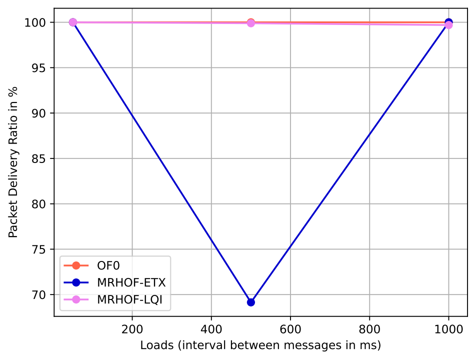


Figure 5.11: PDR of OF0, MRHOF-ETX and MRHOF-LQI for loads of 1 s, 500 ms and 50 ms packet intervals



(a) PDR of uplink for different packet intervals



(b) PDR of downlink for different packet intervals

Figure 5.12: PDR of uplink (left) and downlink (right) OF0, MRHOF-ETX and MRHOF-LQI for loads of 1 s, 500 ms and 50 ms packet intervals.

Fig. 5.13 shows the CDF of per node PDR at a 500 ms packet interval. The graph shows for every possible PDR value on the x-axis, the fraction of nodes that achieved that PDR. On the figure OF0 and MRHOF-LQI have a steep ascend that stays narrow which indicates all nodes achieved a similarly high PDR. Meanwhile, the graph of MRHOF-ETX has a wide, gradual rise. This indicates some nodes achieved a high PDR of up to 90% while some achieved a considerably lower PDR than the rest, resulting in the average PDR of about 70% displayed in Fig. 5.11.

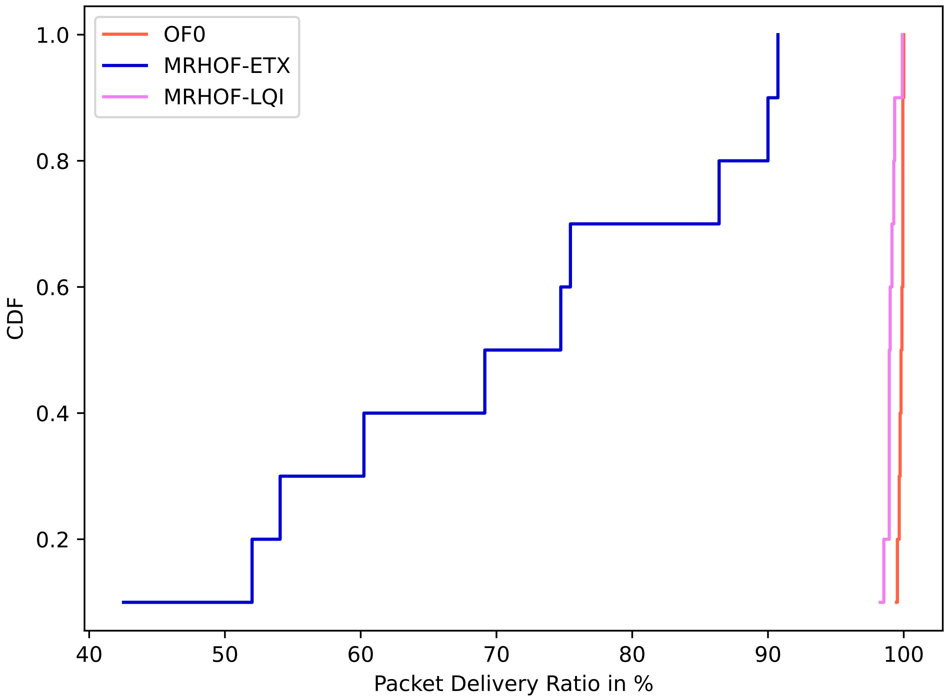


Figure 5.13: CDF diagram of OF0, MRHOF-ETX and MRHOF-LQI for a loads of a 500 ms interval

Fig. 5.14 shows the PDR of OF0 and MRHOF-Energy for a load of a one second interval for the experiment conducted at the University of Applied Sciences Hamburg. The two OFs achieved a similarly high PDR with small differences.

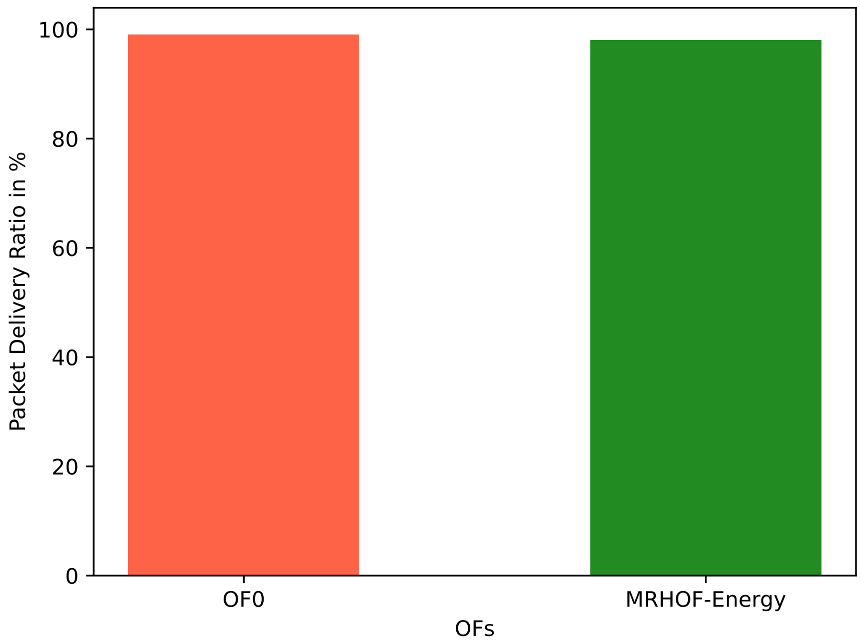


Figure 5.14: PDR of OF0 and MRHOF-Energy

5.2.4 Link Quality

Link quality metrics describe the quality of individual wireless links between neighboring nodes, rather than directly measuring overall network performance as metrics like PDR or RTT do. Nodes with consistently strong link quality are desirable, because they indicate a stable, reliable connection that requires few retransmissions and indicates less packet loss. As these metrics influence how an OF selects parents, but do not give much indication about network performance by themselves, we will examine their relationship to PDR instead.

The ETX [30] metric is the number of transmissions a node expects to make in order for a packet to successfully arrive at its destination. In the best case a packet only has to be transmitted once. However, if the packet gets lost in transit, the packet has to be retransmitted. The retransmission is repeated until the package arrives safely at its destination or the predefined number of maximum retransmissions is reached. Every new retransmission requires energy and time. Hence, a low ETX is desired for a link.

In Fig. 5.15 the PDR of a node is shown on the y-axis and the average ETX value of the link of a node to its preferred parent on the x-axis. If the ETX was directly influencing the PDR, nodes with a low ETX would have a high PDR and nodes with a high PDR would have a low PDR. This pattern can be vaguely seen in Fig. 5.15 with data points scattered in a diagonal line from the upper left to the lower right corner. However, there are also some outlier nodes from MRHOF-ETX and MRHOF-LQI which have an ETX value pf 4.7 to 5.0 but still performed worse than other nodes with the same ETX value.

The LQI [32] metric describes the strength and quality of a received packet. The best possible LQI value is 255 and the worst is 0.

Fig. 5.16 shows the average LQI value on the x-axis and the PDR of a node on the y-axis. If LQI directly influenced PDR, points would be scattered in a line from the bottom left to the upper right corner. The graph shows a gradual, more curved ascend than a straight line. There are also few outliers with LQI values of 140 to 160 which have a relatively low PDR, compared to nodes with a worse LQI but a higher PDR value.

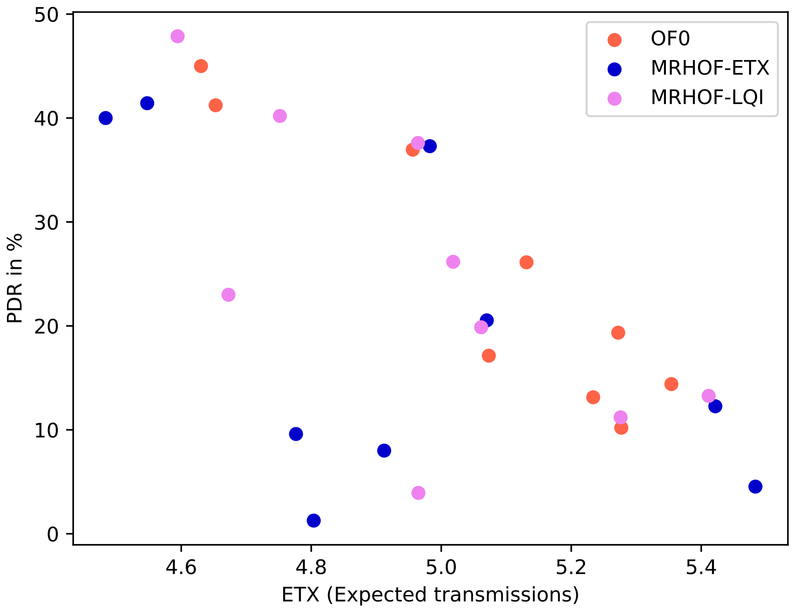


Figure 5.15: Relationship between ETX and PDR at 50 ms packet interval

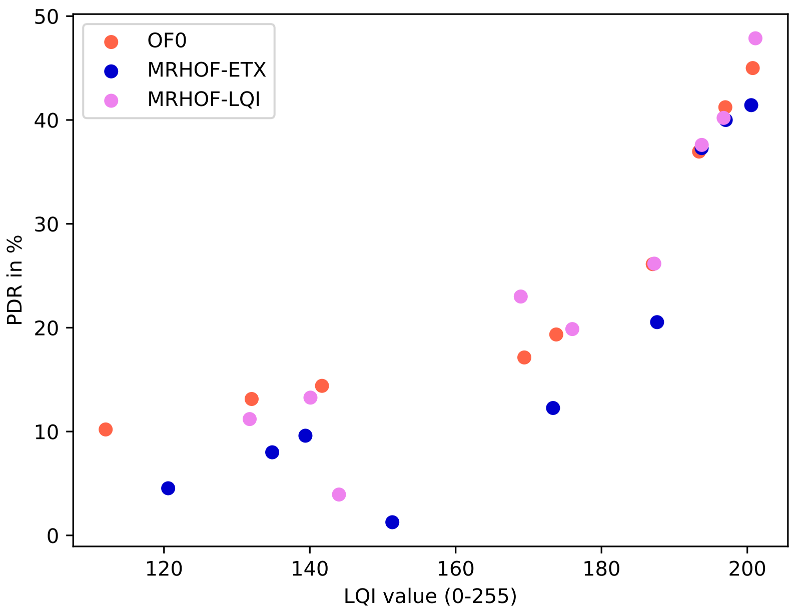


Figure 5.16: Relationship between LQI and PDR at 50 ms packet interval

5.2.5 Round Trip Time

The RTT is the time it takes for a packet to travel to its destination and back. A short RTT is desirable for a network because it indicates that data can be exchanged between nodes with minimal delay. This is especially important for time-sensitive applications, where a delayed response can reduce safe operations of the application.

A consistently low RTT also suggests a stable, efficient routing path with few retransmissions and delays. The cause of a high RTT may be congestion or packets waiting to be processed in a queue while the node is busy forwarding earlier packets. Packets can even be dropped fully due to the queue being full.

Another cause may be packet collisions. When many packets are sent in a short amount of time and arrive at a receiving node at the same time, radio signals of nodes can interfere with each other at the receiver. In that case packets can be lost and have to be retransmitted, which results in a higher RTT.

Fig. 5.17 shows the average RTT in µs of all nodes for the loads of packets being sent in a one second, 500 ms and 50 ms interval. In general, RTT gets prolonged as the interval of packets being sent grows shorter.

Regardless of the interval between messages, OF0 always achieved the shortest RTT. Meanwhile, MRHOF-LQI always produced the second shortest RTT, with its RTT on average being about 7000 µs longer than that of OF0. At last MRHOF-ETX always had the longest RTT. Its gap to the OF0 RTT also kept widening as the interval between messages grew shorter. At a 1 s interval, the gap was about 15000 µs. At a 500 ms interval, the gap was double of that, about 30000 µs, and at the shortest time interval of 50 ms, it was about 25000 µs.

This is however to be expected in a small network of only ten nodes. While OF, as well as MRHOF-LQI, for loads of shorter packet intervals, produced a circular topology where almost all nodes chose the root node as their parent and thus sent their packets directly to its destination without any stations in between, MRHOF-ETX produced a more tree-like topology where the packages of most nodes had to travel more than one hop to reach the root node. Each hop is one more place where congestion, packet drop and retransmission can happen and prolongs the travel time of a packet.

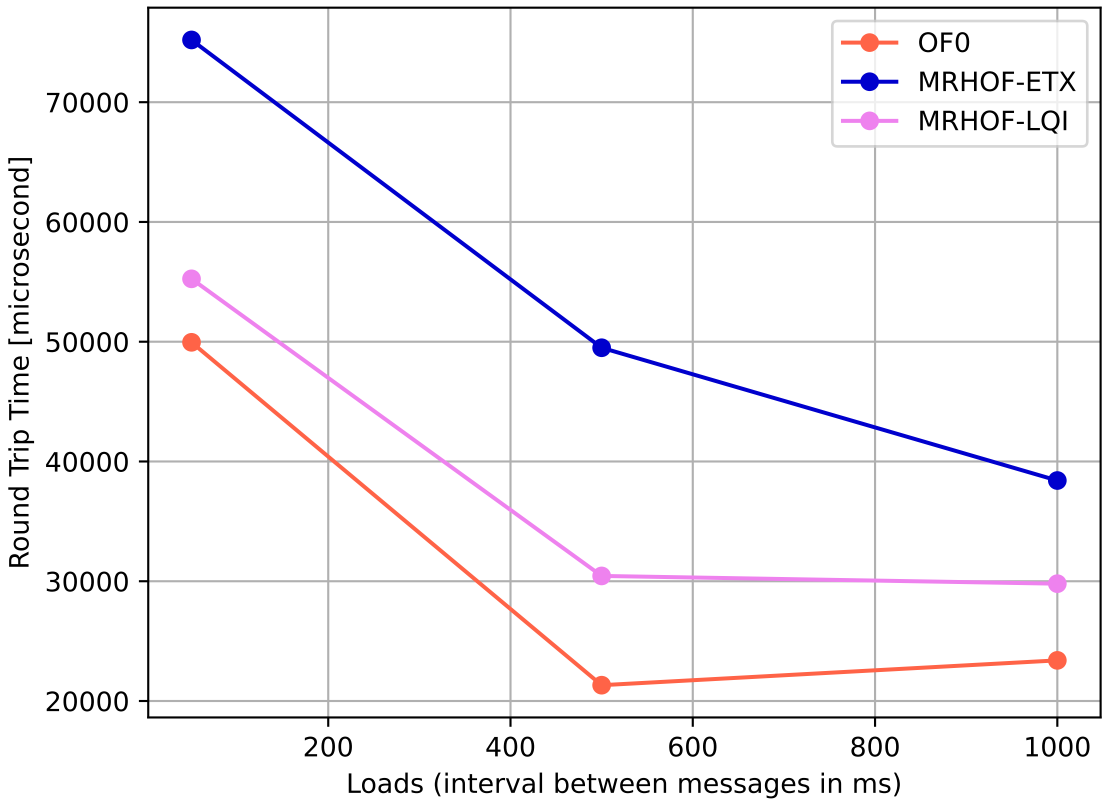


Figure 5.17: Average RTT of OF0, MRHOF-ETX and MRHOF-LQI for package load of 1 s interval, 500 ms interval and 50 ms interval

Fig. 5.18 shows the CDF of RTT for OF0, MRHOF-ETX and MRHOF-LQI at a load of a 1 s interval between packages. The CDF shows the fraction of packets that were delivered within that RTT for every possible RTT value. The graph shows a distribution of RTT values across the network rather than just a single average value. A RTT value can only be computed if a ping message was responded to by a corresponding pong message. So the curve of an OF only included ping messages which were also answered by a corresponding pong message. The ping message might have not arrived, the pong message might not have been sent, or the pong message might have been lost during transit. If any of these cases occurred, the curve does not reach the fraction of 1.0 at any point.

MRHOF-ETX only reached a maximum fraction of 0.93, meaning only 93% of all ping messages successfully received a pong message back. OF0 and MRHOF-LQI meanwhile reached a higher maximum fraction of 0.98 and 0.99.

All three OFs have a minimum RTT in the range of 10.897 to 10.918 µs. The curve of OF0 has a steep, almost vertical ascent, reaching the 0.9 fraction at 34.640 µs and the 1.0 fraction at 93.542 µs. MRHOF-LQI has a steep ascent as well but at a slightly wider angle. Its RTT values are distributed at slightly larger values. It reaches the 0.9 fraction at 45.236 µs, only a difference of 10.596 µs. MRHOF-ETX only reached the 0.9 fraction at 73.330 µs, twice the value of OF0.

Fig. 5.19 shows the CDF of RTT for OF0, MRHOF-ETX and MRHOF-LQI at a load of a 500 ms and 50 ms interval between packets. Compared to Fig. 5.18, in Fig. 5.19a, the difference in recorded RTT values between MRHOF-ETX and the two other OFs is even more apparent. The minimum RTT of all three OFs is in the range of 10.821 µs to 10.933 µs, which is about the same range as before. And while the maximum RTT of OF0 and MRHOF-LQI even shortened, the maximum RTT value of MRHOF-ETX grew by about 50.000 µs.

In Fig. 5.19b RTT values grew longer for all OFs and all curves reached their peak at a longer RTT. One noticeable change is that while MRHOF-ETX reached a maximum RTT of 256.683 µs before with a packet interval of 500 ms, its highest value in Fig. 5.19b is lower at 190.037 µs. This, however, may be due to PDR dropping and packets being lost instead of retransmitted many times and thus reaching their destination but with a long RTT.

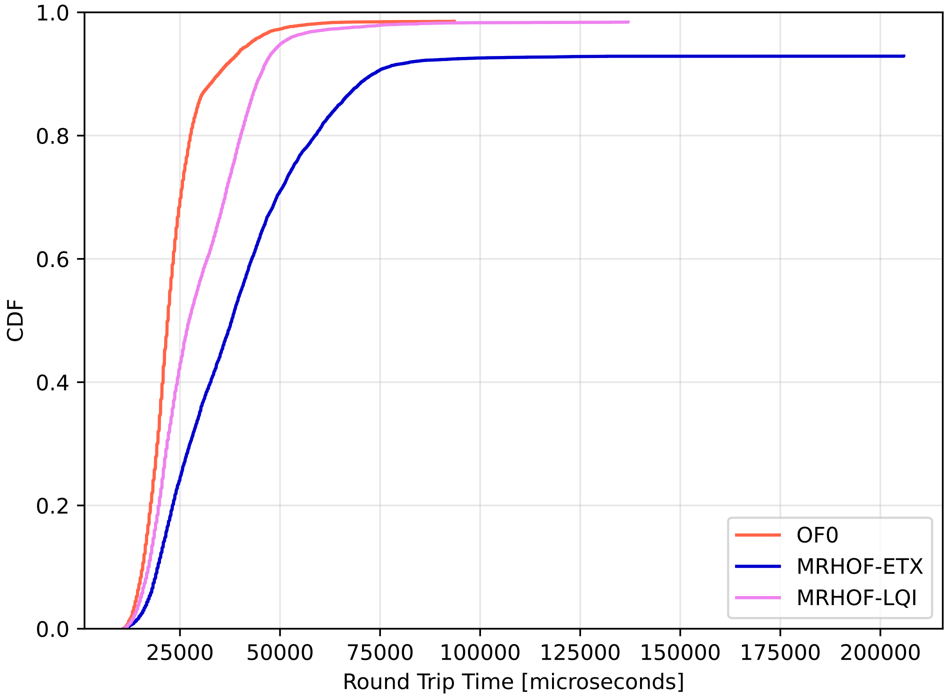
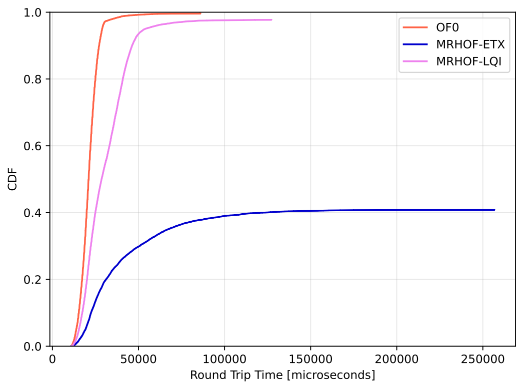
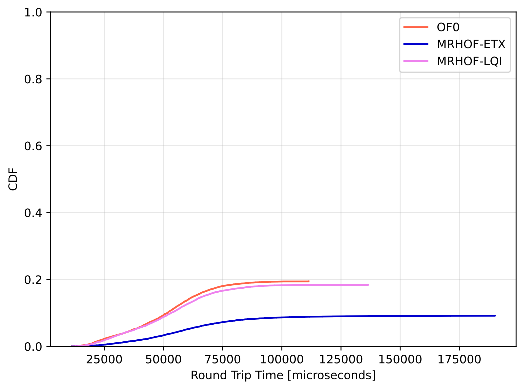


Figure 5.18: CDF of the RTT of OF0, MRHOF-ETX and MRHOF-LQI for package load of 1 s interval



(a) CDF for package load of 500 ms interval



(b) CDF for package load of 50 ms interval

Figure 5.19: CDF of the RTT of OF0, MRHOF-ETX and MRHOF-LQI for package loads of 500 ms (left) and 50 ms (right) interval

5.2.6 Energy

We will now discuss the energy data recorded during the experiments at the University of Applied Sciences Hamburg using lithium polymer batteries. This part will be divided into three different metrics: the energy consumption rate, the energy distribution and the estimated network lifetime.

Node 2 is not included in the results as it was the root node and was permanently connected to a constant power source in order to collect experiment data and observe the experiment as the experiment was running.

Energy Consumption Rate

The energy consumption rate, also known as power, indicates how much energy is consumed by a node in a given amount of time. Usually, this metric is measured in watt or kilowatt. During the experiment we recorded the voltage values of each node. However, there is no data sheet available for the battery used for our experiments. So we cannot convert our voltage measurements into an actual power value. As such our results will use the difference in voltage values for energy consumption.

Fig. 5.20 shows the average energy consumption rate under OF0 and MRHOF-Energy during the experiment. The graph displays MRHOF-Energy as the more energy consuming OF per node. Under both OFs, nodes 1, 4, 5 and 6 consumed considerably more energy than other nodes. Additionally to sending and routing packets, they had LEDs on their board turned on. Under MRHOF-Energy node 6 consumed a little more energy than the nodes 1, 4 and 5. This node was, especially in the beginning, chosen as a parent by nodes 8, 9 and 10 and had the additional task of routing their packets to the root node. Under OF0, this task was taken over by node 5.

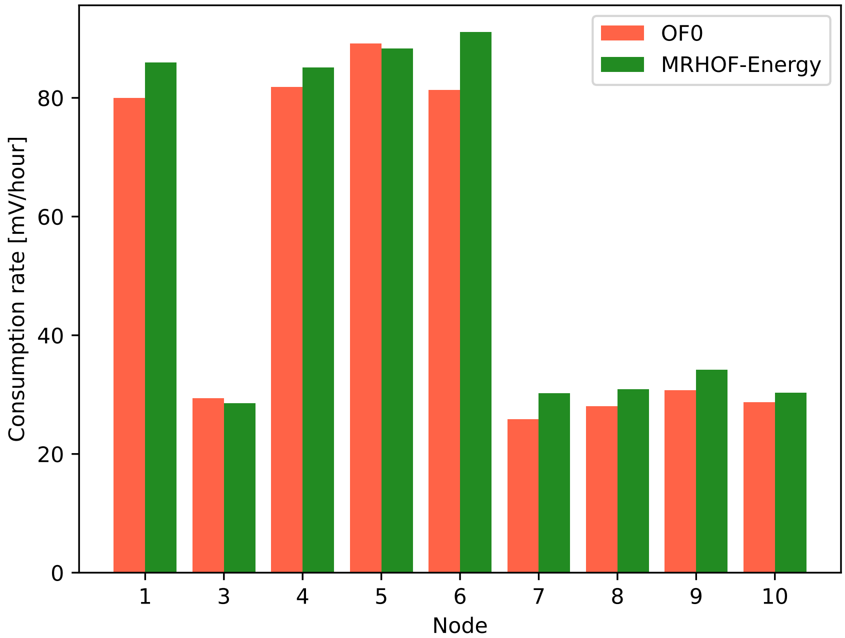


Figure 5.20: Energy consumption rate per node of OF0 and MRHOF-Energy in mV/h

Energy Distribution

The energy distribution refers to how remaining battery capacity is spread across individual nodes in a network. It indicates whether energy consumption is shared evenly among nodes. In general, remaining battery levels of the same values or similar values are an advantage to a network as it extends the network lifetime, the time until the first node of a network dies or a certain amount of nodes in a network have died.

Fig. 5.21 shows the energy distribution at the end of the experiment for every node for OF0 and MRHOF-Energy. All batteries were fully charged at the beginning of the experiment. Nodes 1, 4, 5 and 6, which had LEDs turned on have visibly less energy remaining. Each node has a similar energy level under all OFs with small differences apparent in each node.

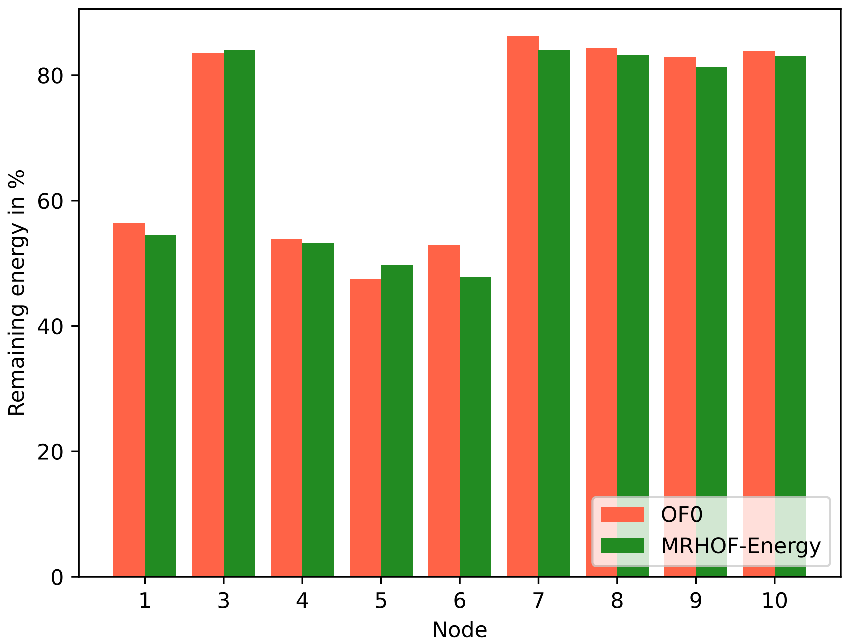


Figure 5.21: Energy distribution per node of OF0 and MRHOF-Energy in %

Estimated Network Lifetime

The lifetime of a network is defined as the time until the first node in a network has died or a specified amount of nodes in the network have died. In this thesis network lifetime is defined as the time when the first node in a network has died. As no node had fully depleted its battery by the end of the experiment, network lifetime was estimated. One approach would have been linear extrapolation, which uses the observed energy consumption rate to estimate values outside the known domain by extending a linear function beyond the existing data. We decided on a similar approach: we extended the data beyond the current data by combining the observed energy consumption rate of each node with the shape of the discharge curve in Fig. 4.4. This enabled a more realistic estimate of the energy discharge of each battery.

The cutoff voltage at which a node was simulated to have depleted its battery was 3500 mV. This value was chosen based on the discharge curve in Fig. 4.4, which shows a steady descent to approximately 3700 mV, before dropping sharply. 3500 mV was selected as it marks this rapid decline, where almost no usable energy remains and the node behavior becomes unreliable, rather than representing the absolute physical depletion point of the battery.

Fig. 5.22 shows the estimated network lifetime of OF0 and MRHOF-Energy in hours. The graph shows the number of nodes still alive after a specific number of hours have passed. The network lifetime achieved under both OFs is close, the first node dying after seven hours and 24 minutes. There is only a small difference between MRHOF-Energy and OF0.

Under OF0 the first node to die is node 5, as it is one of the nodes which have LEDs turned on. Moreover, it was chosen as the parent of nodes 6, 8, 9 ,and 10 throughout the experiment and often had to route their packets additionally to its own.

Under MRHOF-Energy, it was node 6 which also had LEDs turned on and acted as a router for nodes 8, 9, and 10 during the experiment.

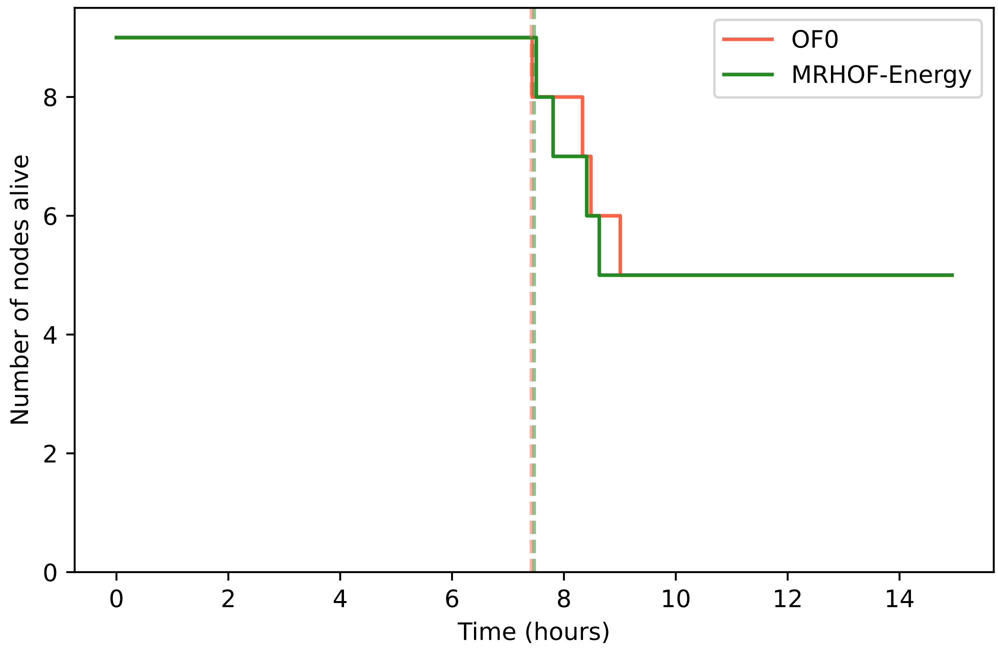


Figure 5.22: Estimated network lifetime of OF0 and MRHOF-Energy in hours

5.2.7 Summary of Results

We have now compared different OFs in their topology, stability of the network, PDR, link quality, RTT, and energy.

The topology created by each OF differs in its shape and amount of hops. OF0 commonly creates a circular shape which minimizes the amount of hops to the root. Nodes usually choose the root node as their parent as long as it is within the range they can send to. In large networks, this leads to one router routing packets for many nodes simultaneously.

MRHOF-ETX and MRHOF-LQI usually create a tree-like topology that branches out with each router having only few children, compared to OF0. However, MRHOF-LQI also created a more circular shape here because the nodes were near to each other and hence had similar LQI values. The circular shape proved to be more advantageous in this small network.

MRHOF-Energy, compared to OF0, had nodes choose routers with more remaining energy as their parent.

When observing only the number of parent changes within a network, the networks under OF0 were the most stable with the fewest parent changes. The most unstable networks operated under MRHOF-ETX with nodes changing their parent quite often.

The PDR of a network can be influenced by different factors, such as the topology, link quality, or congestion. OF0 and MRHOF-ETX were successful on a similar level when it came to PDR. MRHOF-ETX also held up at a 1 s packet interval but lacked, compared to others when packets came in faster intervals.

While inspecting the relationship between the link quality metrics ETX and LQI and the performance metric PDR, a connection could be observed. A strong link often came hand in hand with a high PDR. However, there were some outliers which proved that other factors still had a large influence on PDR and a strong link is not the only prerequisite to a high PDR.

The shortest RTT was always achieved by OF0, seconded by MRHOF-LQI. The longest RTT had MRHOF-ETX. This can be attributed to the multiple hops a packet often had to make in MRHOF-LQI and especially MRHOF-ETX networks. In a small network, a direct connection to the root node proved to be the fastest route.

Initially, OF0 was seen to use less energy than glsac:mrhof-Energy. When it came to network lifetime, however, MRHOF-Energy was estimated to last longer by a small margin. Although, these values were only estimated from existing values. The discharge curve of a battery can vary depending on the tasks executed by the board.

6 Conclusion

6.1 Summary

This work aimed to add an alternative OF to the RIOT OS [3] RPL module which until now only offered OF0 for use. The added OF was to offer RIOT OS [3] users the possibility to choose a OF which focused on routing metrics other than HC, offering a variety of metrics on which the OF could base its routing decisions.

We implemented MRHOF with the metrics ETX, LQI and an energy metric to choose from. The desired metric for routing can easily be switched using the modularity which RIOT OS [3] builds on. The implementation included multiple tasks, such as older implementation inspections, the decision of routing metrics, and the research of boards. One of many challenges was the lack of boards which supported the RIOT OS [3] module ETX implementation. Another was the implementation of an energy metric and its needed hardware components.

Afterwards, we conducted various experiments and analyzed their results to test the functionality of our implementation. The first experiment was conducted on the Fit IoT-LAB [1] with the same nodes but differing packet loads. The second experiment was conducted at the University of Applied Sciences Hamburg using batteries attched to nodes. However, the experiments we were able to conduct were limited due to hardware availability.

We observed that while our MRHOF implementation utilizing the LQI metric could keep up with OF0 in terms of PDR, its effectiveness in terms of RTT was slightly lower than OF0.

Our MRHOF implementaion utilizing the ETX yielded fairly good results but feel behind in terms of PDR and RTT, compared to OF0 and when used with the LQI metric.

In order to improve this shortcoming, adjusting the threshold parameters, used to decide on thresholds when a node will switch to a different parent for MRHOF might lead to better results.

In terms of energy metrics, our MRHOF implementation used with the energy metric yielded a slight improvement in terms of network lifetime, compared to OF0.

Overall, we have now presented an alternative OF to the RIOT OS [3] RPL module. However, the implementation still possesses some shortcomings and requires more testing.

6.2 Future Work

This project included a lot of work and implementation, but some more improvements and extensions can be made from here.

As MRHOF, when used with the ETX metric, was unstable during the experiments, with nodes changing parents frequently and performing worse than OF0 and its other MRHOF variant, some effort can be put into adjusting the MRHOF threshold parameters to stabilize its nodes more.

It would also be favorable to conduct some experiments with MRHOF on larger networks of more than ten nodes in order to observe its behaviour in a different environment.

MRHOF now offers three metrics to choose from. More metrics can easily be added to the implementation to expand the variety MRHOF offers.

In addition to MRHOF, more OFs are in existence that can be implemented. MRHOF offers different routing metrics, and the possibility to easily switch between these routing metrics. But focusing on one routing metric often leaves other constraints which are neglected. It is, however, not possible to make routing decisions based on two routing metrics simultaneously in MRHOF.

Therefore, the next step would be to design and implement a new OF or a modified OF based on MRHOF which takes more than one routing metric or constraint into consideration.

Bibliography

[1] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne, "FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed," Dec. 2015, [Online]. Available: https://hal.inria.fr/hal-01213938. Accessed: Jul. 31, 2026.

[2] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Mar. 2012, [Online]. Available: https://www.rfc-editor.org/info/rfc6550. Accessed: Jul. 19, 2026.

[3] E. Baccelli, C. Gündoğan, O. Hahm, P. Kietzmann, M. S. Lenders, H. Petersen, K. Schleiser, T. C. Schmidt, and M. Wählisch, "Riot: An open source operating system for low-end embedded devices in the iot," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4428–4440, 2018.

[4] C. Bormann, M. Ersue, and A. Keränen, "Terminology for Constrained-Node Networks," RFC 7228, May 2014, [Online]. Available: https://www.rfc-editor.org/info/rfc7228. Accessed: Jul. 16, 2026.

[5] S. R. Das, C. E. Perkins, and E. M. Belding-Royer, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, Jul. 2003, [Online]. Available: https://www.rfc-editor.org/info/rfc3561. Accessed: Jul. 17, 2026.

[6] "Recommendation ITU-T G.9903: Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks," International Telecommunication Union (ITU-T), 2017, annex D specifies the Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng). [Online]. Available: https://www.itu.int/rec/T-REC-G.9903. Accessed: Jul. 19, 2026.

[7] *ZigBee Specification*, ZigBee Alliance, Aug. 2015, defines the Network Layer (NWK) including Cluster Tree and Mesh routing protocols. [Online]. Available: https://csa-iot.org/. Accessed: Jul. 17, 2026.

[8] O. Gnawali, R. Fonseca, K. Jamieson, M. Kazandjieva, D. Moss, and P. Levis, "CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 10, no. 1, pp. 1–49, 2013, [Online]. Available: https://doi.org/10.1145/2529988. Accessed: Jul. 17, 2026.

[9] A. Tavakoli, S. Dawson-Haggerty, J. Hui, and D. Culler, "HYDRO: A Hybrid Routing Protocol for Lossy and Low Power Networks," Internet Engineering Task Force, Internet-Draft draft-tavakoli-hydro-01, Mar. 2009, work in Progress.

[10] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)," RFC 6552, Mar. 2012, [Online]. Available: https://www.rfc-editor.org/info/rfc6552. Accessed: Jul. 16, 2026.

[11] O. Gnawali and P. Levis, "The Minimum Rank with Hysteresis Objective Function," RFC 6719, Sep. 2012, [Online]. Available: https://www.rfc-editor.org/info/rfc6719. Accessed: Jul. 16, 2026.

[12] K. Rembor, *Adafruit Feather nRF52840 Sense*, Adafruit Industries, New York, May 2026, [Online]. Available: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-feather-sense.pdf. Accessed: Jul. 10, 2026.

[13] *nRF52840 DK Hardware*, 026 Nordic Semiconductor ASA, Oslo, January 2026, [Online]. Available: https://docs.nordicsemi.com/r/bundle/ug_nrf52840_dk/page/ug/dk/intro.html. Accessed: Jul. 10, 2026.

[14] FIT IoT-LAB, "IoT-LAB M3," FIT IoT-LAB, 2012, [Online]. Available: https://iot-lab.github.io/docs/boards/iot-lab-m3/. Accessed: Jul. 10, 2026.

[15] I. Kechiche, I. Bousnina, and A. Samet, "A comparative study of RPL objective functions," in *2017 Sixth International Conference on Communications and Networking (ComNet)*, 2017, pp. 1–6.

[16] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," 2004, pp. 455–462.

[17] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, 2006, pp. 641–648.

[18] W. Alayed, L. Mackenzie, and D. Pezaros, "Evaluation of RPL's Single Metric Objective Functions," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, Jun. 2017, p. 154.

[19] N. Gupta, A. Pughat, and V. Sharma, "A critical analysis of RPL objective functions in internet of things paradigm," *Peer-to-Peer Networking and Applications*, vol. 14, no. 4, pp. 2187–2208, 2021, [Online]. Available: https://doi.org/10.1007/s12083-021-01180-9. Accessed: Jul. 13, 2026.

[20] H. Lamaazi and N. Benamar, "A comprehensive survey on enhancements and limitations of the RPL routing protocol," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 45–73, 2017.

[21] S. A. Abdel Hakeem, A. A. Hady, and H. Kim, "RPL routing protocol performance in smart grid applications based wireless sensors: Experimental and simulated analysis," *Electronics*, vol. 8, no. 2, 2019, [Online]. Available: https://www.mdpi.com/2079-9292/8/2/186. Accessed: Jul. 13, 2026.

[22] D. Todoli-Ferrandis, S. Santonja-Climent, V. Sempere-Paya, and J. Silvestre-Blanes, "RPL routing in a real life scenario with an energy efficient objective function," in *2015 23rd Telecommunications Forum Telfor (TELFOR)*. IEEE, Nov. 2015, p. 70.

[23] S. P. Shetty and U. K. K. Shenoy, "Performance of RPL objective functions using fit iot lab," in *International Conference on Innovative Computing and Communications*, D. Gupta, A. Khanna, S. Bhattacharyya, A. E. Hassanien, S. Anand, and A. Jaiswal, Eds. Singapore: Springer Singapore, 2021, pp. 623–630.

[24] S. B. Shetty and M. Shetty, "Energy efficient RPL objective function using fit iotlab," in *International Conference on Innovative Computing and Communications*, D. Gupta, A. Khanna, A. E. Hassanien, S. Anand, and A. Jaiswal, Eds. Singapore: Springer Nature Singapore, 2023, pp. 229–238.

[25] P. O. Kamgueu, E. Nataf, T. Djotio Ndié, and O. Festor, "Energy-based routing metric for RPL," INRIA, Research Report RR-8208, Jan. 2013, [Online]. Available: https://inria.hal.science/hal-00779519. Accessed: May. 07, 2026.

[26] L.-H. Chang, T.-H. Lee, S.-J. Chen, and C.-Y. Liao, "Energy-efficient oriented routing algorithm in wireless sensor networks," in *Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics*, ser. SMC '13. USA: IEEE Computer Society, 2013, p. 3813–3818, [Online]. Available: https://doi.org/10.1109/SMC.2013.651. Accessed: Jul. 16, 2026.

[27] J. Nurmio, E. Nigussie, and C. Poellabauer, "Equalizing Energy Distribution in Sensor Nodes through Optimization of RPL," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Perv2015cit..confR*. IEEE, Oct. 2015, pp. 83–91.

[28] O. Iova, F. Theoleyre, and T. Noel, "Using Multiparent Routing in RPL to Increase the Stability and the Lifetime of the Network," *Ad Hoc Networks*, vol. 29, Jun. 2015, [Online]. Available: https://hal.science/hal-01206380. Accessed: Jul. 16, 2026.

[29] G. Sun, X. Zheng, J. Li, H. Kang, and S. Liang, "Collaborative wsn-uav data collection in smart agriculture: A bi-objective optimization scheme," *ACM Trans. Sen. Netw.*, May 2023, just Accepted.

[30] D. Barthel, J. Vasseur, K. Pister, M. Kim, and N. Dejean, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks," RFC 6551, Mar. 2012, [Online]. Available: https://www.rfc-editor.org/info/rfc6551. Accessed: Jul. 29 2026.

[31] I. A. Alnajjar, "A comprehensive survey on objective functions in RPL routing with various networking and application scenarios," *EAI Endorsed Transactions on Internet of Things*, vol. 11, no. 1, 9 2025.

[32] *IEEE Standard for Low-Rate Wireless Networks*, Institute of Electrical and Electronics Engineers, 2024, iEEE Std 802.15.4-2024 (Revision of IEEE Std 802.15.4-2020).

[33] B. Valentin. gnrc/rpl: Implement minimum rank with hysteresis objective function. RIOT. GitHub Pull Request #15623. [Online]. Available: https://github.com/RIOT-OS/RIOT/pull/14623. Accessed: Jul. 10, 2026.

[34] B. Valentin, B. Stottelaar, and C. Büchse. UDP benchmark server. RIOT. [Online]. Available: https://github.com/RIOT-OS/RIOT/tree/85b7ff0a0079a062a1fbdb6b2bf1d6960f3d927f/dist/tools/benchmark_udp. Accessed: Jul. 20, 2026.

[35] B. Valentin, M. Buschsieweke, fzi-haxel, and E. Frank. UDP benchmark client. RIOT. [Online]. Available: https://github.com/RIOT-OS/RIOT/tree/1ce183495a3bbe5a31a025beddaa8b7deb44cf7d/sys/test_utils/benchmark_udp. Accessed: Jul. 20, 2026.

Erklärung zur selbständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit in allen Teilen selbstständig angefertigt und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt habe. Die in meiner Arbeit verwendeten KI-basierten Hilfsmittel habe ich (ggf. mit Produktnamen) angegeben.

Ich verantworte die Übernahme jeglicher von mir verwendeter maschinell generierter Passagen vollumfänglich selbst und trage die Verantwortung für eventuell durch die KI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Mir ist bewusst, dass wahrheitswidrige Angaben als Täuschungsversuch behandelt werden können.

Ort	Datum	Unterschrift im Original