Routing in Internet of Things (IoT)

By

Sammia Rauf

MSCSF16M025

Supervised by

Dr. Shahzad Sarwar

Assistant Professor, PUCIT

(Session 2016-2018)

Punjab University College of Information Technology,

University of the Punjab, Lahore, Pakistan.

Routing in Internet of Things (IoT)

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF PHILOSOPHY

IN

COMPUTER SCIENCE

By

Sammia Rauf

MSCSF16M025

Supervised by

Dr. Shahzad Sarwar

Assistant Professor, PUCIT

Co - Supervised by

Dr. Laeeq Aslam

Assistant Professor, PUCIT

(Session 2016-2018)

Punjab University College of Information Technology,

University of the Punjab, Lahore, Pakistan.

Evaluation of M. Phil. Thesis

We have evaluated the M. Phil. thesis titled **Routing in Internet of Things (IoT)** Submitted by Ms. Sammia Rauf, roll no. MSCSF16M025, session 2016-2018 in partial fulfillment of the M. Phil. degree in Computer Science. We have also assessed the candidate through viva-voce.

We are satisfied with the thesis and performance of the candidate in the examination and are of the opinion that he fulfills the requirements as set in the rules and regulations for the M. Phil. degree in Computer Science at the University of the Punjab.

Thesis Supervisor:

Dr. Shahzad Sarwar
Assistant Professor
Punjab University College of Information Technology
University of the Punjab, Lahore

External Examiner:

Dr. Ghulam Mustafa
Assistant Professor
School of Systems and Technology
University of Management and Technology, Lahore

Principal of the College:

Major (R) Muhammad Arif Butt
Principal,
Punjab University College of Information Technology
University of the Punjab, Lahore

UNIVERSITY OF THE PUNJAB

Author: **Sammia Rauf**

Title: **Routing in Internet of Things (IoT)**

Department: **Punjab University College of Information Technology**

Degree: **M. Phil. (Computer Science)**

Signature of the Author

Dedicated to

my family...

Abstract

Internet of Things (IoT) demands the global connectivity for even a small tiny object that can interact with its environment, can communicate and identifiable as long as possible. While Low Power and Lossy area Networks (LLNs) have some constraints Like they are very small in size, have small and low power batteries, wireless, limited capacity of storing, communication and computation. These LLNs are also the part of IoT. Because of such constrained environment and to fulfill the global connectivity, lightweight protocols are preferred for IoT. Routing is an important part of inter-networking. Traditional routing protocols of TCP/IP stack have a large overhead in IoT solutions. That is why, there is need for a light weight protocol for routing also for IoT. Routing Protocol for Low Power and Lossy Area Networks (RPL) was proposed by an Internet Engineering Task Force (IETF) working group of Routing over Low Power and Lossy Area Networks (ROLL). RPL has some deficiencies and does not provide any load balancing mechanism for a node. RPL can be improved terms of objective function, hysteresis and trickle timer also. In this research, we have focused on RPL a routing protocol in IoT systems. After understanding of the RPL, its existing solution is simulated on a suitable simulator, which help us to identify the problems in this protocol. After that we have found an improved solution for RPL and made its comparison with existing protocol, which eventually help us to improve the performance of IoT.

Keywords: Internet of Things (IoT), low power and lossy area networks (LLNs), routing protocol, RPL.

Acknowledgments

Firstly, I would thank to Almighty Allah who enabled me to work for a cause. He gave me courage to withhold the difficult circumstances. He enabled me to show my strengths regarding my subject area to others and to be a part of an enlightened arena.

Secondly, I would like to thank my thesis advisor Dr. Shahzad Sarwar and co advisor Dr. Laeeq Aslam who worked diligently in the compilation of my research work. I have no words to pay gratitude to them. Both of them played a supportive role whenever I ran into a trouble spot or had a query about my research work. They always helped me towards the right direction.

In the end, but foremost I am thankful to my parents, family and friends who supported me in the hardships. Their guidance and supervision have made me feel blessed in order to complete my thesis. May Allah Almighty bless them all…

Table of Content

List of Figures

List of Tables

Chapter 1 Introduction

Internet of Things is in its evolving stages, there exists many ways to define IoT. In literature there exist a lot of definitions for “Internet of Things” here are the most relevant: In paper [1] authors have defined the IoT: “An open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting in face of situations and changes in the environment”. In [2] Authors have said that: “A conceptual framework that leverages on the availability of heterogeneous devices and interconnection solutions, as well as augmented physical objects providing a shared information base on global scale, to support the design of applications involving at the same virtual level both people and representations of objects”. IoT can also be defined as an idea for which “IoT states that various “things”, including a tiny physical object on the planet, are going to be connected and will be controlled across the Internet” [3]. Our simplest definition of IoT is: “Accessing and control “Things” using “Wireless Networks” over “The Internet”, is called “Internet of Things””.

Gartner report in 2013 said that: "The growth in IoT will far exceed that of other connected devices. By 2020, the number of smartphones tablets and PCs in use will reach about 7.3 billion units," said Peter Middleton, research director at Gartner [4]. "In contrast, the IoT will have expanded at a much faster rate, resulting in a population of about 26 billion units at that time". As we mentioned earlier that in IoT the term “Things” can be anything from a small tiny object to a large huge object and a living person as well. Because of heterogenous and constrained devices, it is very difficult to implement existing solutions in IoT.

There exist many algorithms for routing in conventional Internet as well as in mobile networks but they are not suitable for IoT based solutions. Because, these algorithms are more complex and required high computational power and other resources. Now let’s on move to the Low Powered and Lossy Area Networks (LLNs) which sometimes also refers to the Wireless Sensor Networks (WSN). These kinds of networks have some characteristics like participating nodes are tiny in size, limited battery power, wireless, limited capacity of storage, communication, and computation.

IoT is based on these kinds of Low Power and Lossy Area Networks (LLNs). Conventional TCP/IP Stack with complicated algorithms is not suitable for these kinds of networks. Although conventional routing algorithms can find a routing path for these networks but they are more complex and will consume more battery power, storage and computation. Which will reduce the overall lifespan of a node and in the worst case we will not be able to communicate with any of the node in that network.

Because of these kinds of issues, we will work on the issue of routing in the Internet of Things. A lot of solutions exist to improve the performance of LLNs Like application protocols for constrained environments COAP [5] which is a substitute of HTTP. Further, low-power

wireless personal area networks (6LoWPAN) is a protocol which supports the fragmentation and reassembly of internet protocol version 6 (IPv6) packets in order to enable encapsulation of IPv6 packets into 802.15.4 frames [13]. This 6LoWPAN protocol is working as an adaptation layer to reduce the packet overhead.

Although, these protocols are improving the performance of LLNs but there is a need to improve the routing mechanism for these networks. To address this issue IETF working group for routing ROLL proposed routing protocol for low power and lossy area networks (RPL) [8].

In this study, our emphasis on the RPL in constrained environments. After a complete understanding about the working of RPL, the existing RPL protocols are being simulated. Furthermore, the shortcomings of the solutions of RPL are being highlighted. In the last stage, we put our efforts to find out an optimal solution which eventually improve the performance of these kind of constrained networks for the Internet of Things.

Chapter 2 covers the related work in RPL and routing in IoT. Chapter 3 explains the proposed solution and simulation setup. Experiments and results are presented in the Chapter 4. In the last the conclusions and future work of thesis is presented.

Chapter 2 Related Work

The research community has decided to introduce lightweight protocols to overcome the issues of constrained environment. So that, they can be easily configured even on small tiny objects. Now, the current protocol stack for IoT is very different from conventional TCP/IP stack as shown in figure 1.

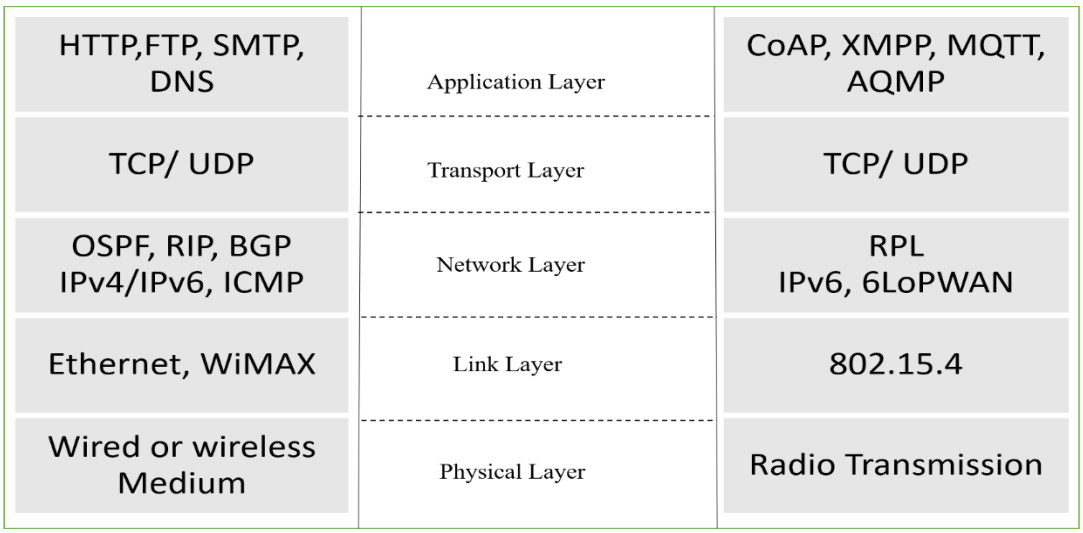

Figure 1: TCP/IP Stack Vs IoT Sack.

At application layer there are many protocols that are being used in IoT Like extensible messaging and presence protocol (XMPP), constrained application protocol (CoAP), message queuing telemetry transport (MQTT), advanced message queuing protocol (AMQP) and hypertext transfer protocol (HTTP). CoAP is the most popular and versatile application layer protocol, because it uses user datagram protocol (UDP) at transport layer. It can emulate transmission control protocol (TCP) like behavior using confirmable messages to meet the application requirement [6]. At network layer IoT devices used IPv6, 6LoWPAN and RPL. 6LoWPAN acts as an adaptation layer and is used to fragmentation of IPv6 packets so that these small packets can be encapsulated in LLNs frames. When each fragmented packet reaches at its destination, 6LoWPAN again reassemble these packets.

2.1. Internet protocol version 6 and low-power wireless personal area networks (6LoWPAN)

6LoWPAN uses four types of header: (i) dispatch header, (ii) mesh header, (iii) first fragment header, and subsequent fragment header. These headers are used to distinguish between point to point small packets, fragmented packets and mesh transmitted packets [13].

2.1.1. Dispatch header

Dispatch header is 8 bits long (Figure 2a). First two bits are 00 or 01 to indicate dispatch header while the next 6 bits are user to distinguished between the header type to follow next in this packet. “000001” for uncompressed IPv6 and “000010” for HCI IPv6 compressed encoding.

2.1.2. Mesh header

Mesh header is identified by “10” (Figure 2b). Mesh header is 4 bytes long and deals with the encoding of hop count and link layer source and destination of packets.

2.1.3. Fragmentation headers

Size of fragmentation header in 4 bytes if it is a first fragment header while 5 bytes for each subsequent fragment header (Figure 2c and 2d). Fragmentation header is mostly used when a large packet is come to a node because 802.15.4 can support the payload up to 102 bytes for a frame.

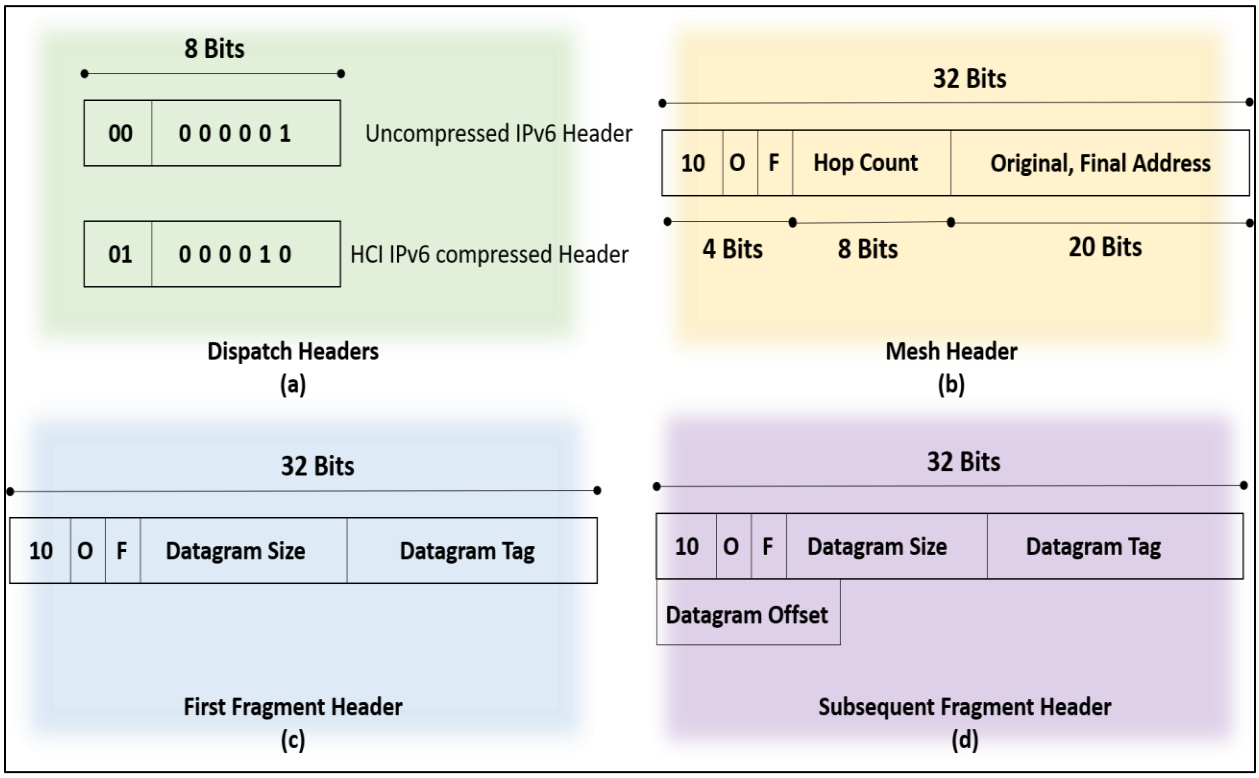

Figure 2: 6LoWPAN Headers Types.

These headers can be used in different combinations to represent each transmitted packet in wireless network. Figure 3 is showing these stacked headers formats. For a small point to point communication only IPv6 compressed header is placed above the UDP header. Furthermore, 6LoWPAN provides header compression scheme for UDP, TCP can be used with 6LoWPAN,

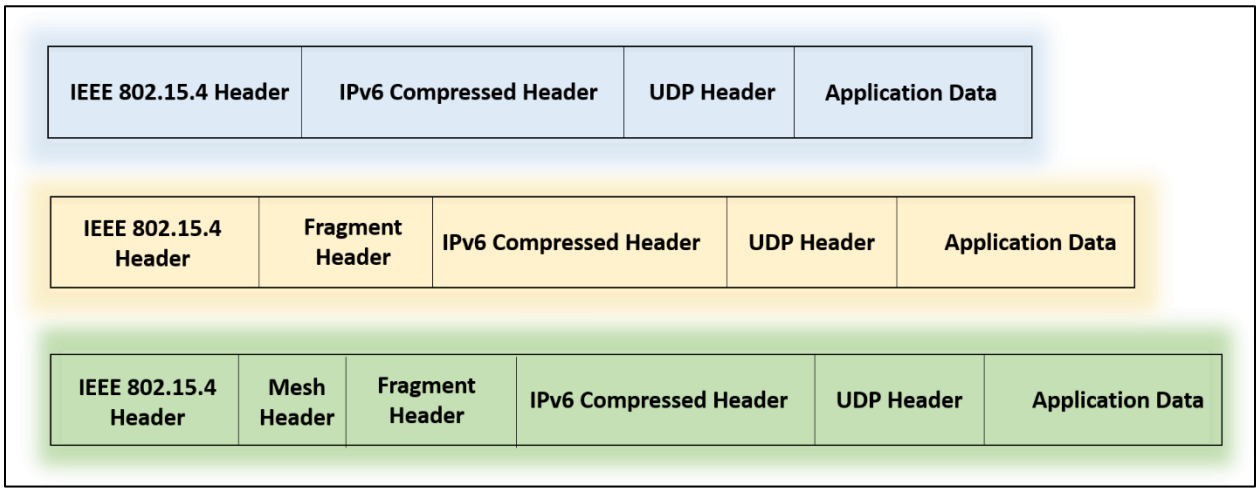

Figure 3: 6LoWPAN Stacked Headers.

however 6LoWPAN does not provides any compression scheme or encoding for TCP header [15]. IETF working group for 6LoWPAN is currently working on TCP header encoding as well but they did not standardize these encoding yet [16] [19]. As mentioned above fragment header is used to break down a large packet. This fragment header stacked along with IPv6 compressed header and also with mesh header for more than one hop forwarding. Mesh header is used to forward data to more than one hop with in a 6LoWPAN network.

2.2. Routing Protocol for Low Power and Lossy areas networks (RPL)

RPL is used at network layer for the routing in LLNs. RPL is distance vector protocol which creates a destination oriented directed acyclic graph DODAG with less complexity. Distance vector is such an algorithm which is widely being used in WSNs for routing. The advantage of distance vector protocol is that pre-knowledge about network topology is not require. A wireless node can become the part of existing network by passing the control messages. The purpose of distance vector is to find the minimum distance between the nodes by maintaining a vector table and communicate it with other nodes [17].

2.2.1. Network Graph

RPL is also a source routing protocol, which means each node discover all possible ways to reach the host to send a packet in the network. Figure 4(a) is showing the nodes in a network. To communicate with each other and message passing these nodes need to connect with other using some common mechanism. While getting connection in a network there is a need to avoid to stuck in loop or cycle, so that a message always reaches to its destination.

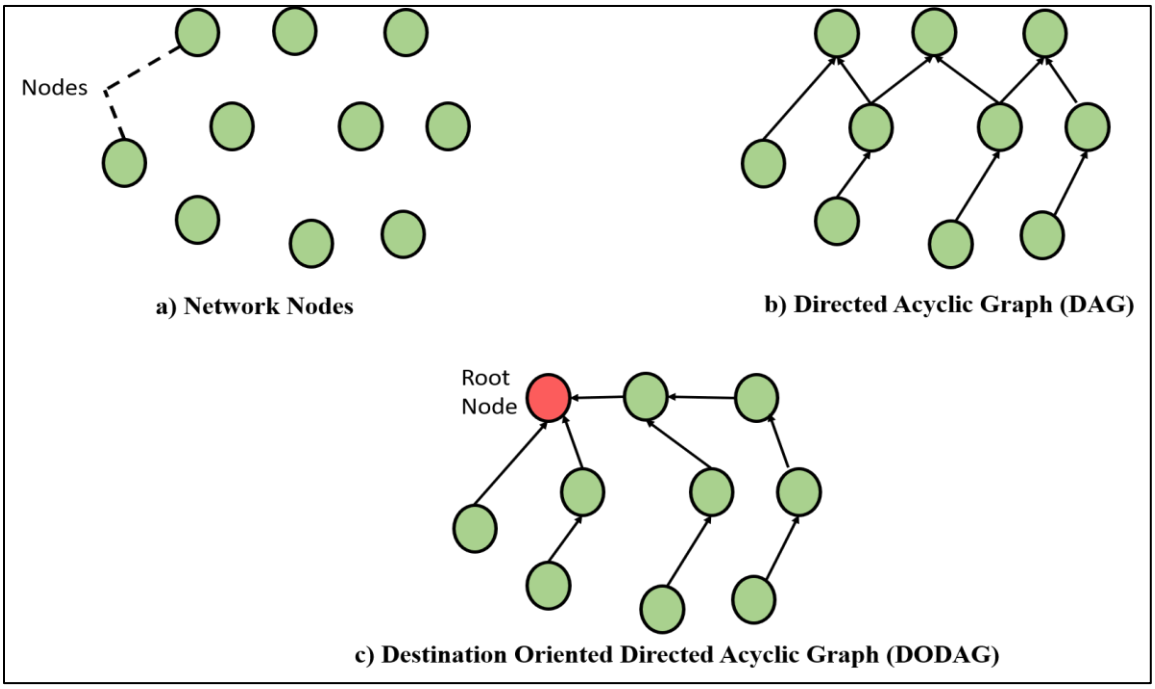

Figure 4: Network Graph.

There is a possibility of a cycle while using distance vector protocol. Which is also called count to infinity problem [18]. This problem can be resolved by making directed acyclic graph as shown in figure 4(b). RPL also creates a DAG pointing toward a destination or source node which is also called a root node in a topology. This root node usually, a live powered node which is acts as a gate way node between an IoT network and the internet. This root node gathers the data from the sink nodes in its region, aggregate it, and send it over the internet to the cloud data centers.

2.2.2. RPL Instances

Each DODAG is identified by an ID which is referred as DODAGID. Figure 5 is showing an RPL instance. Each RPL instance can have one or more DODAGs with same RPL instance ID but different DODAGIDs. Each RPL instance have same objective function for all of its DODAGs.

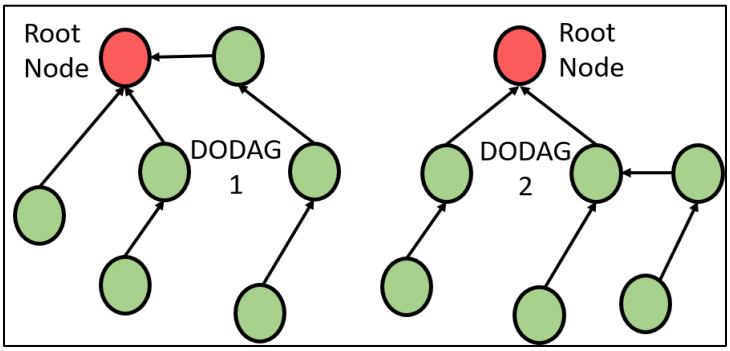

Figure 5: RPL Instance.

We will discuss objective function later in this chapter. Anyhow objective function is used to calculate the rank of node with in a DODAG.

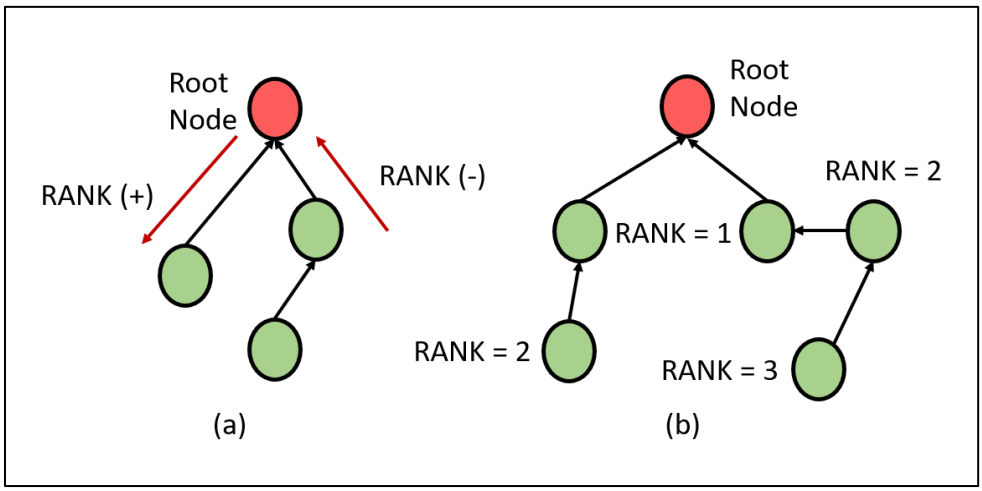

Figure 6 : DODAG Nodes Rank.

Rank defines the position of a node relative to the other nodes with in a DODAG with respect to root node. Root node has the least rank, which makes it superior among other nodes. While a leaf node has an infinity rank until it becomes the part of a DODAG. Figure 6 is demonstrating the node rank with in RPL DODAG. Rank will always increase downwards to the root node and decrease towards the root node. Each node wants to connect with a parent node having the least rank among other potential parent nodes.

2.2.3. RPL DODAG Types, Storage and Traffic

2.2.3.1. DODAG Categories

DODAGs are further categorized into two types:

i. Grounded DODAG.
ii. Floating DODAG.

When every node in a DODAG provide a connectivity towards the root node is called grounded DODAG. While the floating DODAG is a disjoint part of a DODAG which does not provide any connectivity to its child nodes towards the root node. Floating DODAG can occur during the RPL repair operation.

2.2.3.2. RPL Memory Mode

RPL has two types of mode for memory:

i. Storing mode.
ii. Non-storing mode.

Each node maintains a routing table of downwards traffic path in storing mode. Packets travel only as far as common parent. Nodes having lower ranks may have bigger tables, that is why storing mode is limited by size of routing table otherwise memory problems will occur and protocol fails when any table is full. While in non-storing mode root node has the routing table in most of the cases and all traffic is sent to the root node and then redirect towards the respective node.

2.2.3.3. RPL Traffic Scenario

RPL can support three types of traffic scenario:

i. Point to multipoint (P2MP).
ii. Multipoint to point (MP2P).
iii. Point to point (P2P).

P2MP traffic travels downward from root node to child node. MP2P traffic travels upward from child node towards root node. While P2P traffic flows between the node in a DODAG. RPL always keep control information messages separate form data messages.

2.2.4. RPL control messages

RPL uses ICMPv6 protocol for sending control information. Figure 7 is showing the format of RPL ICMPv6 messages. To indicate RPL message ICMPv6 type is 155. Code field indicate the type of RPL control message. In RPL there are total four type of control messages.

i. DODAG information solicitation (DIS).
ii. DODAG information object (DIO).
iii. Destination advertisement object (DAO).
iv. Destination advertisement object acknowledgement (DAO-ACK).

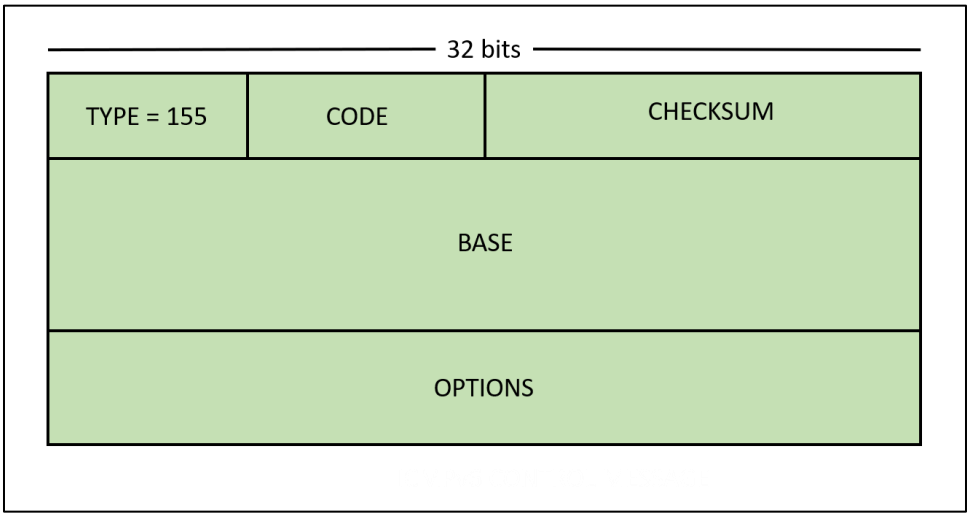

Figure 7 : RPL ICMPv6 Message.

Code for DIS is 0x00, DIO is 0x01, DAO is 0x02 and DAO-ACK is 0x03. Base field of ICMP message contains these control messages in the form of an object. While options field contains optional data if any.

2.2.4.1. DODAG information solicitation (DIS)

Now we will discuss RPL control messages in detail. Figure 8 is showing the format of DIS object. DODAG information solicitation (DIS) is used to discover devouring nodes so that they can initiate a DIO messages. First 8 bits of DIS object used to set flags and next 8-bit field reserved for future use. While the length of options field can vary, it contains a valid option for

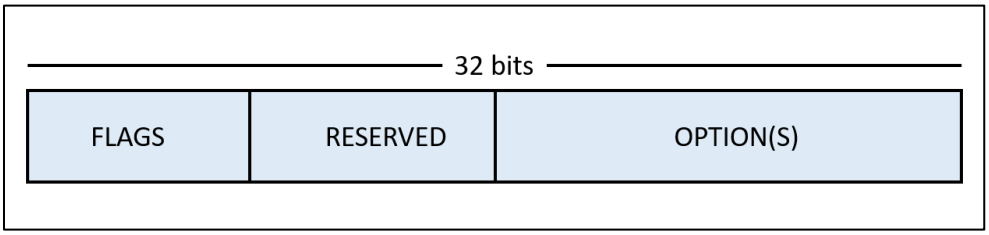

Figure 8: DIS Base Object.

messages 0x07 for solicitation information, 0x00 for PAD1, and 0x01 for PADN [8]. DIS traffic is very high in the beginning of DODAG formation because only root node or child node having parent can send DIO. While a leaf node can send a DIS only until it will find some DIO message from some node.

2.2.4.2. DODAG information object (DIO)

With the help of DIO messages a node can discover an RPL instance to join and disseminate information. Able to find its configuration parameters to a DODAG preferred parents. DIO further helps to maintain the DODAG downward rout.

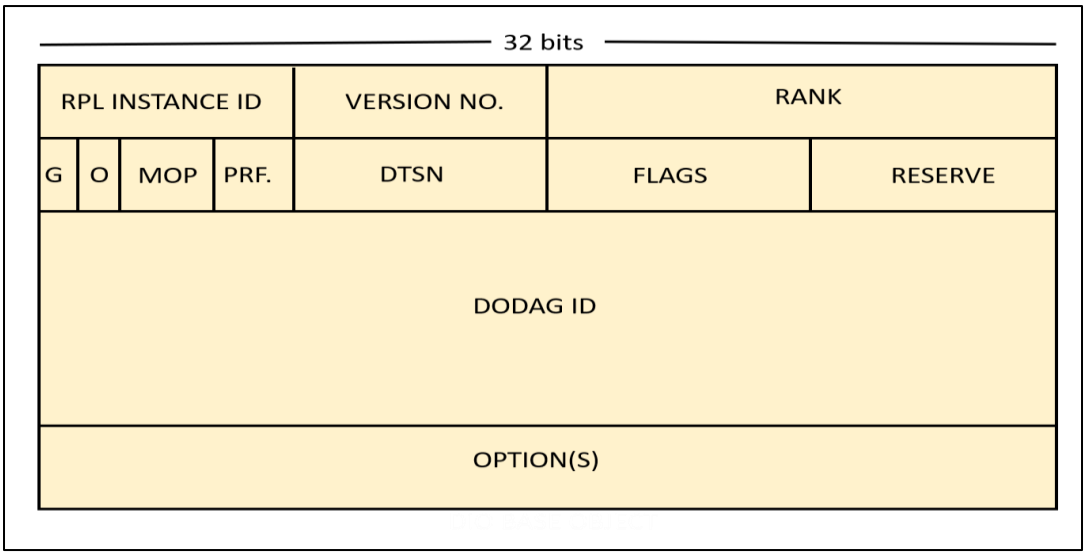

Figure 9: DIO Base Object.

Figure 9 is showing the format of DIO base object. First eight bits are used to show RPL instance id. Next eight bits for DODAG version number. After that rank is placed with 16 bits which means maximum rank of a node can be 65535. G and O flag is showing the nature of DODAG either grounded or floating. Next four bits are for the mode of operations.

Every node in a DODAG have to follow the instructions in MOP field to participate as a router, otherwise it can only participate as a leaf node. Next field is to indicate preferred parent. Next eight-bit field is a Destination Advertisement Trigger Sequence Number (DTSN) that is used to maintain downward routes in a DODAG.

Next 8 bit is for flags which is set to zero if not required. Eight bits are reserved for any future use. Next field is 128 bits long for DODAG id. It is an IPv6 address set by a DODAG root. The DODAGID MUST be a routable IPv6 address belonging to the DODAG root node [8]. While the options field usually contains a metric container.

2.2.4.3. Destination advertisement object (DAO)

Figure 10 is showing a DAO base object format. Just like DIO first 8-bit field indicating the RPLInstanceID. 1 bit 'K' flag indicates that the recipient of this DAO has to send a DAO-ACK back. Next 1 bit 'D' flag indicates that the DODAGID field is present after that 6-bit flags and 8-bit reserved field is present.

Next 8-bit field is for DAO sequence which Incremented at each unique DAO message from a node and echoed in the DAO-ACK message. After that DODAGID and options field is preset just like DIO [8].

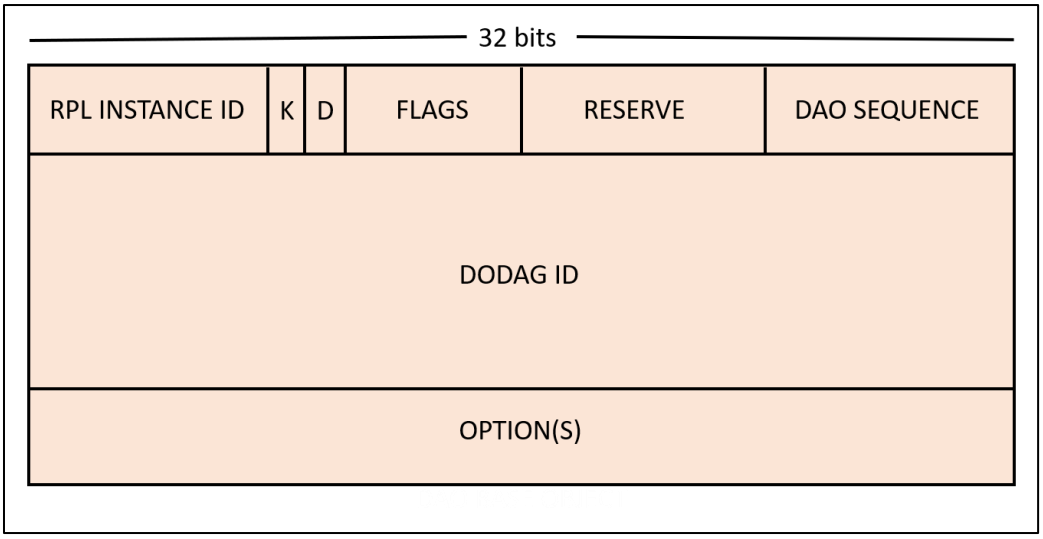

Figure 10: DAO Base Object.

2.2.4.4. DAO acknowledgment base objects

Figure 11 is showing the format of DAO acknowledgment base objects. DAO-ACK base object format is quite similar with DAO base object. Only a status field is different from DAO. 8-bits status field indicates the DAO completion. Status 0 means a DAO is accepted but unqualified [8]. Status from 1 to 127 means rejection but not absolutely, which means the node sending the DAO-ACK can be a parent but not a suitable one and eventually receiving DAO-ACK node has to find some other parent. If status is in between 127 to 255 that means the node sending DAO-ACK rejects to become a parent.

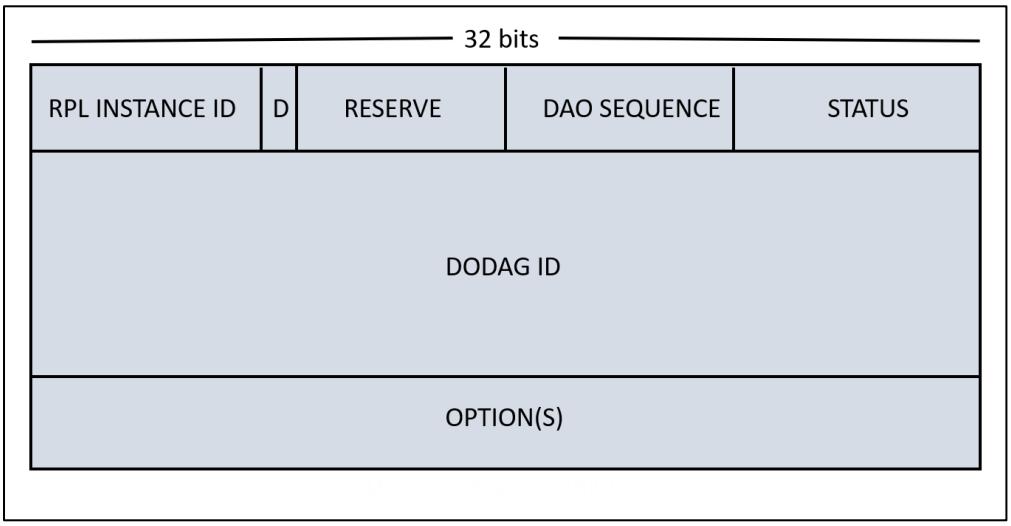

Figure 11: DAO-ACK Base Object.

2.2.5. Metric container

Metric container that is present in a DIO base object used to disseminate metrics or constraints with in a DODAG. On the bases of these metrics a node selects its preferred parent in a DODAG and calculate its rank. Upon receiving a message, each node compares its rank with respect to its parent. Parent selection is based on the objective function and hysteresis [7]. We will discuss these objective functions later in detail. There are two types of constraints node constraints and link constraints [8]. Node constraints or metrics, can be a node state or an attribute object which is for CPU or memory usage. Node energy object is about the energy or power information that a node is consuming. Hop count object depicts the information about the number of hops between root and this node. While link constraints/ metrics can be throughput, latency, and reliability in a low power and lossy area network link reliability that can be measured with expected transmission count (ETX) which actually is an average number of transmissions a node takes to deliver a packet.

2.2.6. Trickle timer

Each node sends a DIO message with respect to trickle timer to the other nodes. It constructs the DODAG topology in such a way that it reduced the control traffic overhead by using:

(i) Suppression.
(ii) Inconsistency state solving.

When a node's neighbors have received sufficient number of transmissions of information, suppression place its role and suppresses the transmission. While the Inconsistency state solving is used whenever a network goes to the inconsistent state. To quickly resolve the inconsistency, the sending rate of control messages in increased by the algorithm [20] [21].

The trickle algorithm has the following three parameters:

(i) Minimum interval size - Imin.
(ii) Maximum interval size - Imax.
(iii) Redundancy constant - k.

Trickle algorithm keeps the record of three variables: (I) Current interval size I, (II) Counter (c), and (III) time during the current interval (t).

The interval boundary of each node is "I" which is greater than "Imin" and less than "Imax". This main interval window is further divided into sub interval window. Each sub interval starts with "Istart = Imin" and ends with 'Iend = Istart*2". At each "Istart" counter (c) is set to zero and time "t" is set randomly in the range of "I/2 to I" by trickle algorithm. Counter "c" is incremented by one each time when a DIO arrived during the first half of interval. In the second half, time (t) is selected to send a DIO message only if c is less than redundancy constant (k). When "I" reaches to "Iend", trickle algorithm increases interval size by a factor of 2. And "I" is set to Imin again whenever an inconsistent state occurs in the network [22] [23].

2.2.7. Objective function

Objective function for an RPL instance state that, how RPL nodes choose and optimize routes within an RPL Instance? Express how nodes translate metrics into a rank. Express how the nodes will select their parents. RPL can further be improved in the terms of objective function hysteresis [11] and trickle timer. Further RPL can be improved by provide any load balancing mechanism for a node in DODAG. RPL has a default objective function (OF) called OF Zero. The goal of this OF is to find the nearest grounded root node. For this Rank is computed for each node which is based on its preferred parent. While parent selection for each node is based on this objective function. In this objective function a single parameter (hop count) is used to compute rank. Instead of the selection of preferred parent, each node maintains a list of successors for backup. In this objective function other node or link parameter can be used as well [14].

There exists another objective function by ROLL working group “Minimum Rank with Hysteresis Objective function (MRHOF)”. This OF is quite same as Objective function Zero. MRHOF suggested to use ETX as a default parameter for parent selection. Parent with the minimum path cost will be selected as a preferred parent. Once a parent is selected for a node, any other candidate parent for that node will not be selected until it’s given parameter is smaller than current parent by certain threshold (Hysteresis) [11].

As we have already known that default RPL is using Link parameter ETX or a node parameter Hop count in its objective function. This paper [12] have presented a new approach for OF parameters. It's a Hybrid approach based on machine learning, in which a weight (theta and gamma) is assigned to the thresholds (hysteresis part) for both the parameters (ETX and Hop count). And a node switches its parent only if candidate parent parameters is greater than from both thresholds. Their results have suggested that the Packet delivery ratio of HYBRID OF is close to ETX based OF when theta is 600 and gamma is 0.8, while number of parent changes, control traffic and energy consumption is also less as compared to the other objective functions.

Both objective functions MRHOF and OF0 rely on a greedy approach for parent selection, which may cause the problem of load balancing. This kind of parent selection can lead to an unbalanced DODAG in the network. To avoid this unbalanced scenario the authors of this draft [9] have proposed a new Load Balancing OF. In this OF a parent will be preferred for a node have a smaller number of child nodes. To fulfill this requirement, they have presented three new modifications in RPL. i) introduce new RPL Metric called CNC (Child Node Count) object, ii) amend preferred parent IP address in DIO message by each node, iii) now parent node accepts a DIO from its child to keep track of its child nodes.

This Load Balancing OF has been further improved by traffic aware load balancing objective function [10]. This objective function has suggested us to add packet transmission rate as a parameter of TAOF to balance traffic load on each node. This is a proposed solution for load balancing not yet implemented or tested. Thus, it is difficult to comment on its significance.

Chapter 3 Methodology

3.1. Theoretical Explanation

We have made three contributions in this research work.

i. Queue length as a metric.
ii. Update Metric Container
iii. Propose an objective function.

First one is we have introduced a new RPL metric which is a queue length. Queue length tell us about the traffic load on each node. This metric will help us to control traffic on each node which eventually avoid congestion with in a network and will reduce buffer overflow. Second, we have updated the structure of metric container. Our metric container contains three different constraints at a time. One is ETX, second one is energy object and third one is queue length. Figure 12 is showing the structure of our metric container.

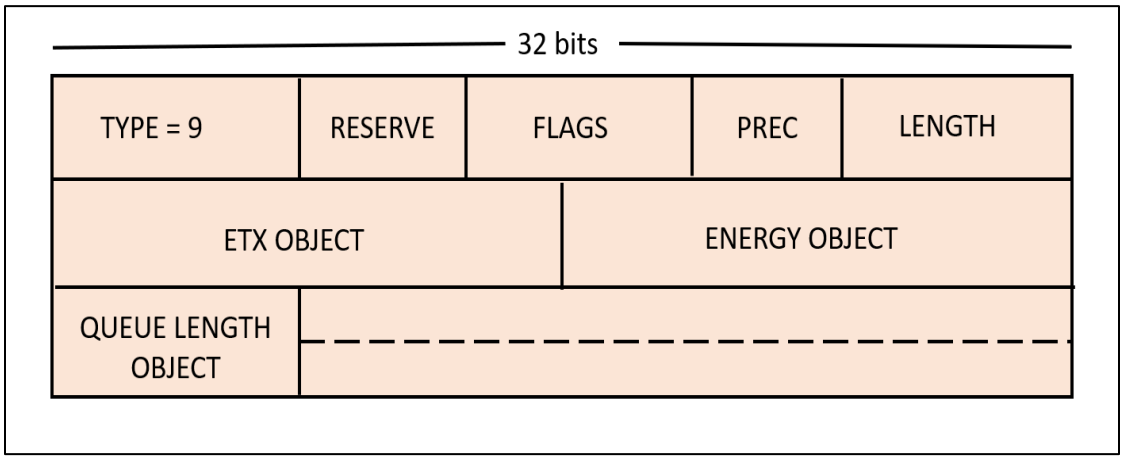

Figure 12: Metric Container.

First 8 bits are for mentioning the type of constraints [25]. Existing constraints/ metrics types are eight so our combine metrics type is 9. Next 5 bits are reserved by IETF for future purposes [25]. After that there are 7 bits for multiple configuration flags and 3 bits for precedence field. 8 bits length field indicate the length of object body in bytes in our case length field is five. In object body we have placed three metrics. ETX object for reporting link quality is 16 bits long. Energy object is also 1 bit long. While the queue length object for reporting the queue size of a node is 8 bits long.

Third we have proposed our objective function (OF) **"SEEMI"**, we have combined three parameters ETX, energy and queue length in our objective function. ETX for link quality, this parameter will help to find out how much good or bad a link is? Energy parameter will control the availability of a mote as long as possible in a network. While our third parameter queue length will

control the traffic load on each node which eventually control the congestion in a network. This objective function can be expressed with the following equation.

$$\boldsymbol{OFSEEMI = \alpha ETX + \beta Energy + \gamma QueueLength} \text{ -------- eq (i)}$$

Here α, β, and γ are the weights for each parameter. The total weightage of α, β, and γ is 1. The values of α, β, and γ is between 0.00 to 1.00 but their sum should be equal to 1.00.

First, we scan all those nodes having a good link quality which means small EXT value, all those links with a bad link quality will be rejected because those nodes have the less probability to transmit packet successfully. Probability of a node to send a packet successfully can be calculate by taking ratio to the total number of attempts.

$$psuccess = 1/(totalnumberofattempts) \text{ -------- eq (ii)}$$

$$psuccess = 1/(totalnumberofsuccess + totalnumberoffailure) \text{ -------- eq (iii)}$$

$$\boldsymbol{psuccess = 1/(1 + ETX)} \text{ -------- eq (iv)}$$

So, we can say that smaller the EXT value the better the link is. After scanning the good EXT links, we can compare all parameters. Now let's talk about our major parameter which is queue length. In LLNs each node except the leaf nodes work as router for other nodes which means they have to forward the packets of other nodes as well as their own application data packets. Size of queue length will determine the work load at each node. Which means nodes with large queue length have high workload and small queue size means less work load at the node. In the case of high work load there is a chance of buffer overflow and packet loss as each node. So, each node in an RPL DODAG will try to pick a node with small queue length which means less workload and smaller the chance of packet loss.

When we were working on above mentioned approach about queue length, we have observed that it is not always true that a node with small queue length having less workload. Smaller queue size could be the result of very good link quality. For example, a node very close to the root node must have a good link quality in such a way that its packet departure rate from queue is greater than or equal to the packet arrival rate.

$$\boldsymbol{packetdeparturerate \geq packetdeparturerate} \text{ -------- eq (v)}$$

So, its queue length will always become zero but it does not mean that node is doing less work in that RPL instance. Because of this reason we have added our third parameter which is the energy consumed by that node. Because we can relate it higher the workload, more the energy will be consumed.

$$\boldsymbol{workload \propto energyconsumption} \text{ -------- eq (v)}$$

Because all of the mentioned reasons, our objective function has three parameters for selecting a suitable parent. Because of three parameters there can be total 2^3 (8) type of categories from which a parent node can belong. We have already mentioned that our OF extracting good EXT links first that is why only four categories will be remaining. Table 1 is showing all those combinations.

OF Parameters	ETX	Energy Consumption	Queue Length
Discarded Parent Types	Bad	High	Large
	Bad	High	Small
	Bad	Low	Large
	Bad	Low	Small
Remaining Parent Types	Good	High	Large
	Good	High	Small
	Good	Low	Large
	Good	Low	Small

Table 1: RPL Parent Types & Combinations.

3.2. Workflow

Root node starts sending DIO messages at first, while the other nodes can only send DIS until thy join the DODAG. When a node receives a DIO, if it is the first time, it accepts the sending node as a parent node otherwise it will compare it with existing rank and then decide whether it has to maintain the same position or has to move in a better one.

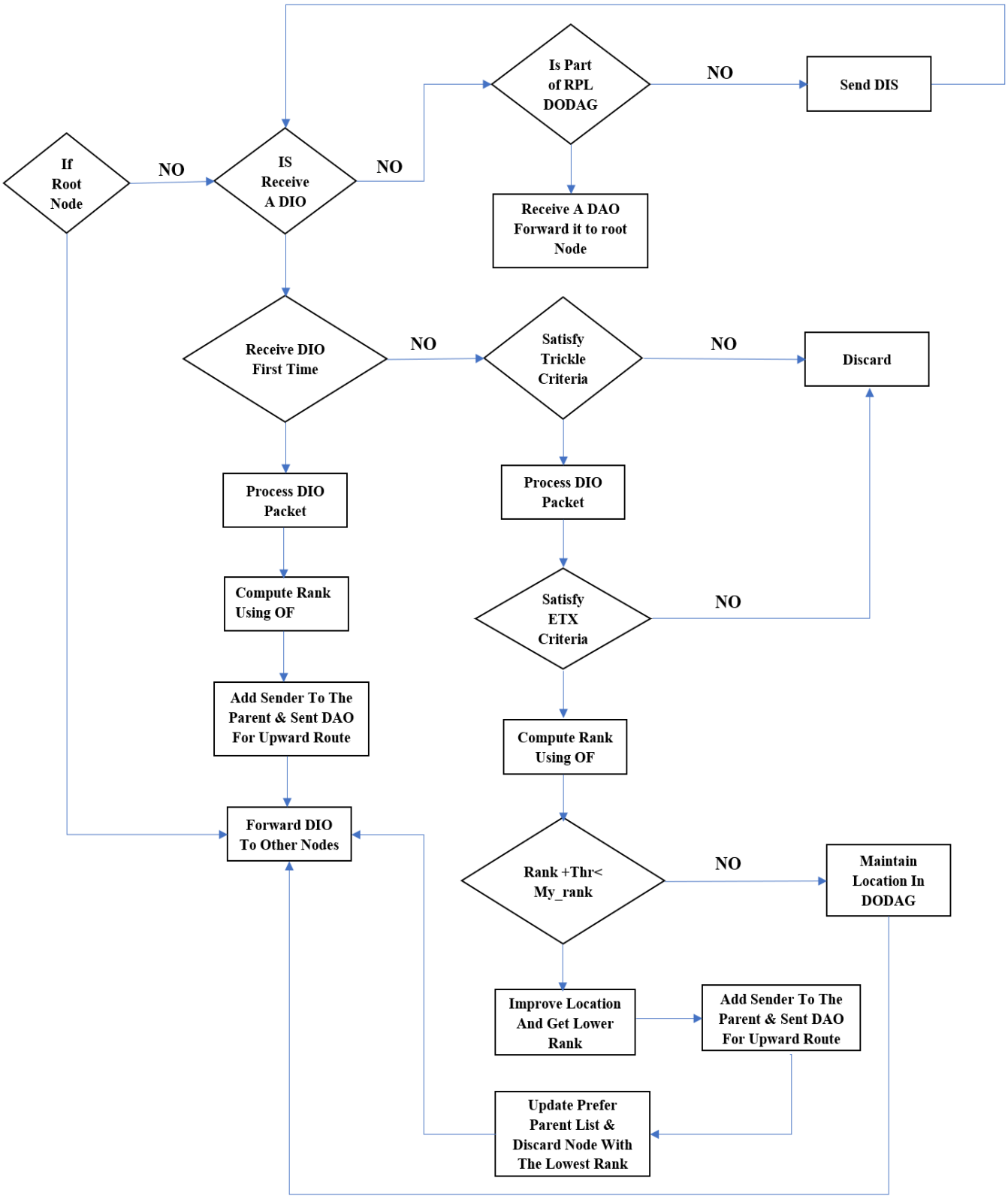

Figure 13: Flow Graph of RPL with Object Function SEEMI.

3.3 Assumptions

1) All equipment, simulator and software are available and free of cost.
2) Simulation results can be different from real infrastructure results.

3.4 Simulation Setup

In this research, to simulate our proposed solution and existing RPL techniques we have used COOJA a Contiki based network simulator [24]. Contiki is an open source operating system for IoT platforms, code for this simulator setup can be found here [26]. In our simulation we have used unit disk graph medium (UDGM) with 1000 milli seconds mote startup delay. We have tested our proposed solution on random topology of multiple nodes with high intensity traffic and low intensity traffic. Figure 14 is showing the working model of COOJA simulator.

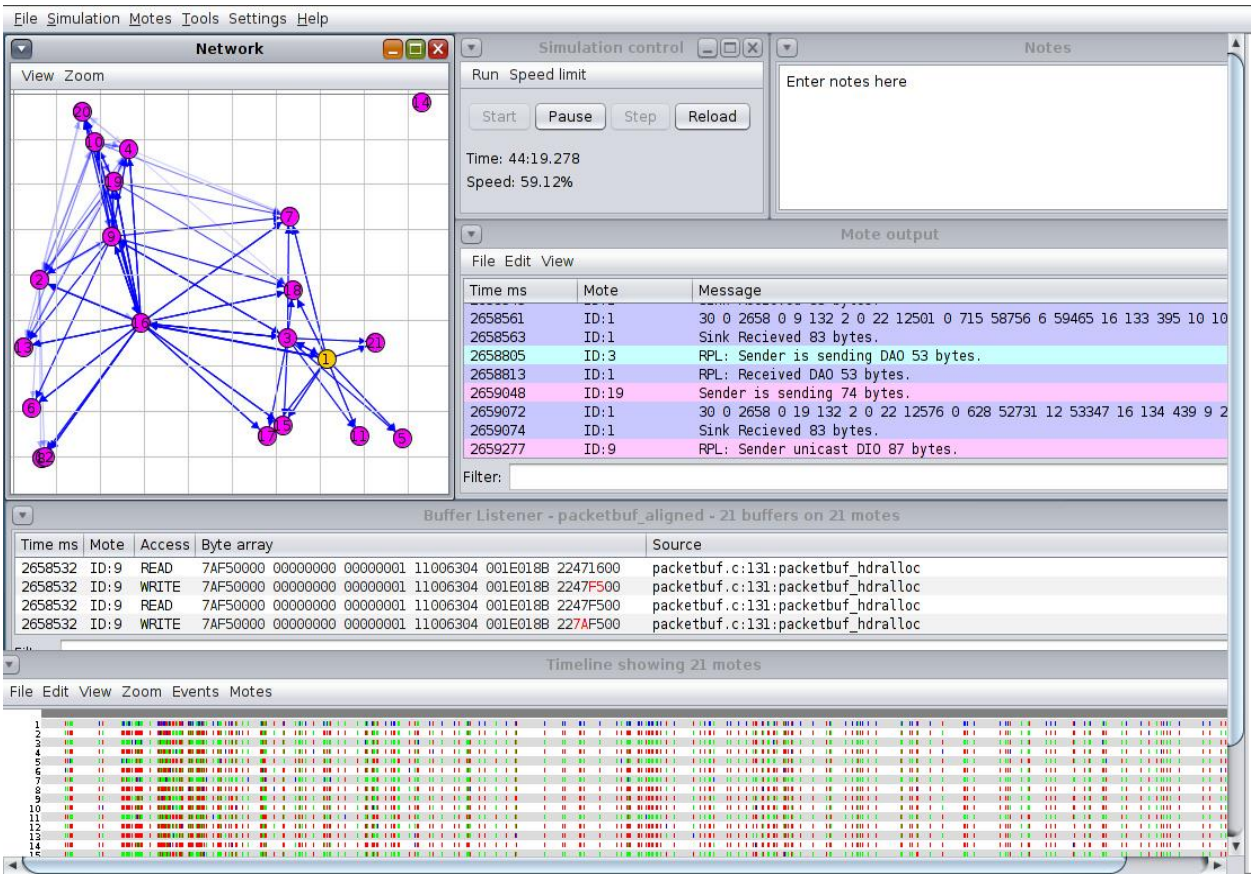

Figure 14: COOJA Simulator Working.

Table 2 is showing the simulation setup that we have used in our experiments. Motes spread area is 100 x 100 m with the transmission range of 25m. We have run our simulation for 30 minutes as per simulation time. This simulation time can be quite different from real time. Which is totally depended on the processing that has to done by simulator and machine as well.

Name	Value
Radio Medium	UGDM
Area	100 x 100 m
Topology	Random
Transmission Range	25 m
Simulation Time	1800s
Simulation Speed Limit	No Limit
Simulation Scenarios	**High Intensity Traffic:** • Time Interval (Seconds): 10, 30, 50 • No. of Motes: 10, 20, 30
	Low Intensity Traffic: • Time Interval (Minutes): 05, 10, 15 • No. of Motes: 40, 60, 30

Table 2: Simulation Setup.

Chapter 4 Experiments & Results

As we have already mentioned, we have performed our simulation on high traffic intensity and low traffic intensity. By doing this our purpose is to cover both real time and non-real time applications. In high traffic intensity our data traffic transmission intervals are 10, 20 and, 30 seconds on 10 motes, 20 motes and 30 motes. While in low traffic intensity our data traffic transmission intervals are 5, 10 and, 15 minutes on 40 motes, 60 motes and 80 motes. We have also compared our objective function with OF0 and MRHOF. Figure 15 is showing the combination or our experimental framework. In our experiments we have compare OF SEEMI

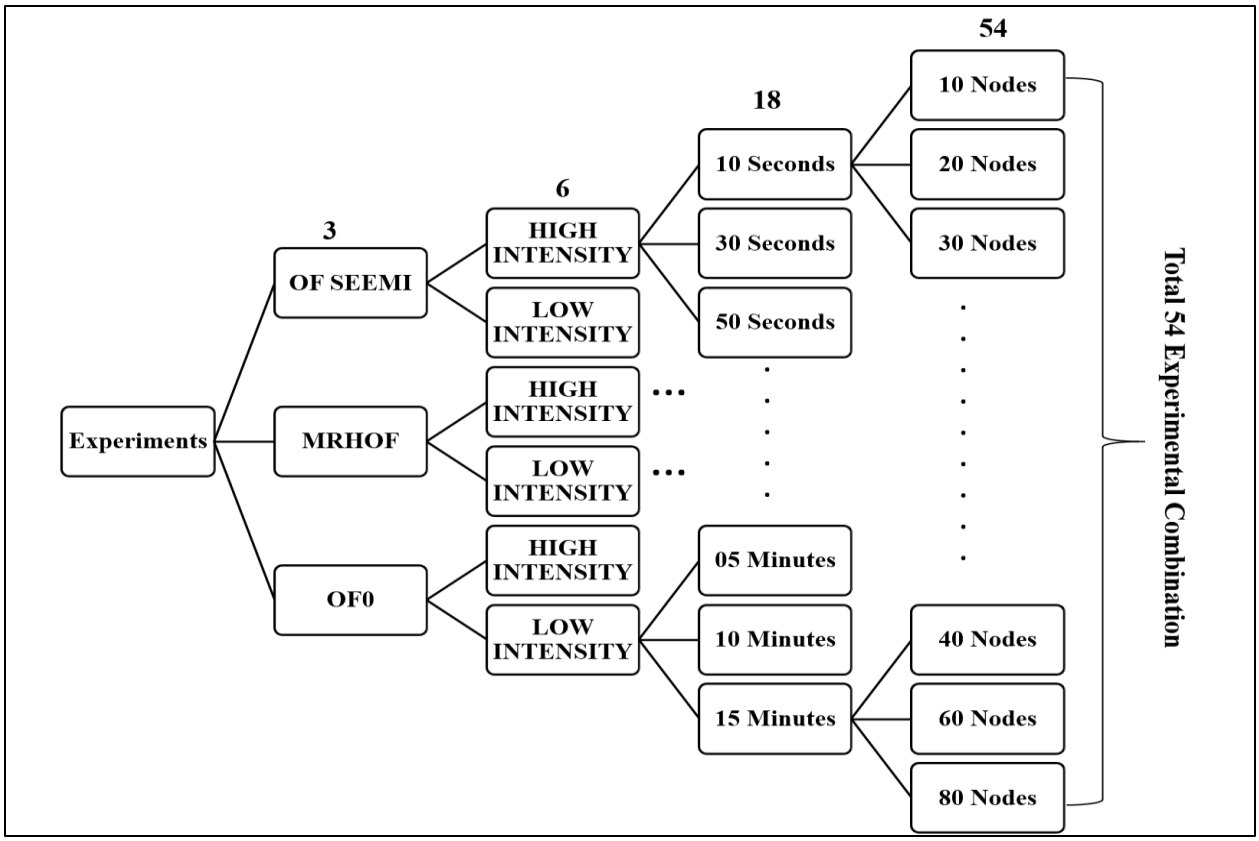

Figure 15: Experimental Framework.

with MRHOF and OF0. We have compared our objective function in terms of control traffic over head and throughput. As we have already mentioned that we have classify our data traffic generation scenarios into two types high intensity traffic and low intensity traffic. we will explain these results according to this classification.

4.1. High Traffic Intensity

In high traffic intensity, we have taken time intervals of 10, 20 and 30 seconds along with 10, 20 and 30 number of motes to compare the results of OF SEEMI with MRHOF and OF0 for real time traffic scenarios. Figure 16, 17, and 18 are showing the results of control traffic over

head. We have computed this overhead in percentage by taking the ratio of control traffic with the sum of control traffic and data traffic.

Our results are showing that our proposed objective function Seemi is performing better than both of the other objective functions in the case of less dense network with 10 motes in terms of control traffic overhead. It is also performing much better than OF0 in more dense networks with 20 and 30 motes. While it is giving almost same control traffic overhead as MRHOF even without hysteresis configurations.

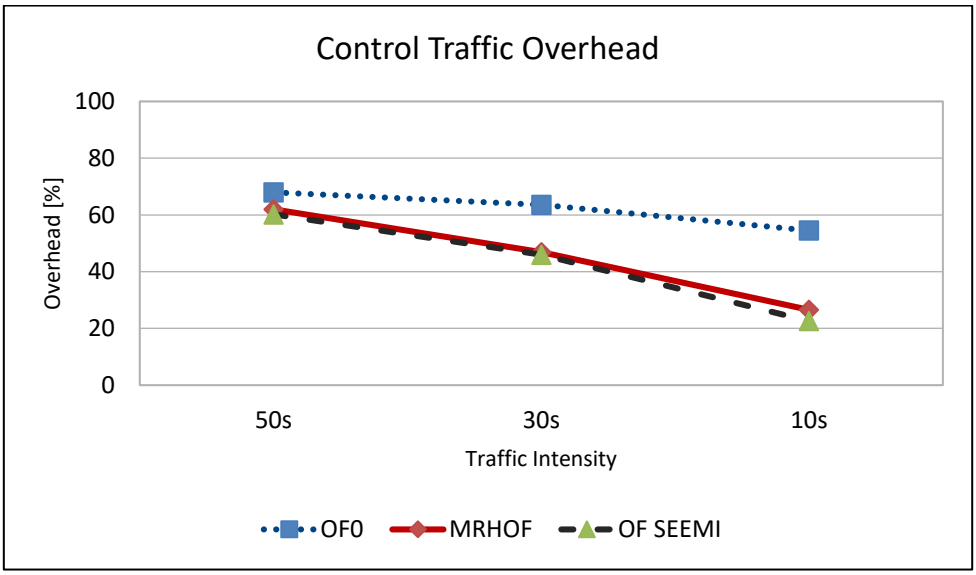

Figure 16: Control Traffic Overhead with 10 nodes.

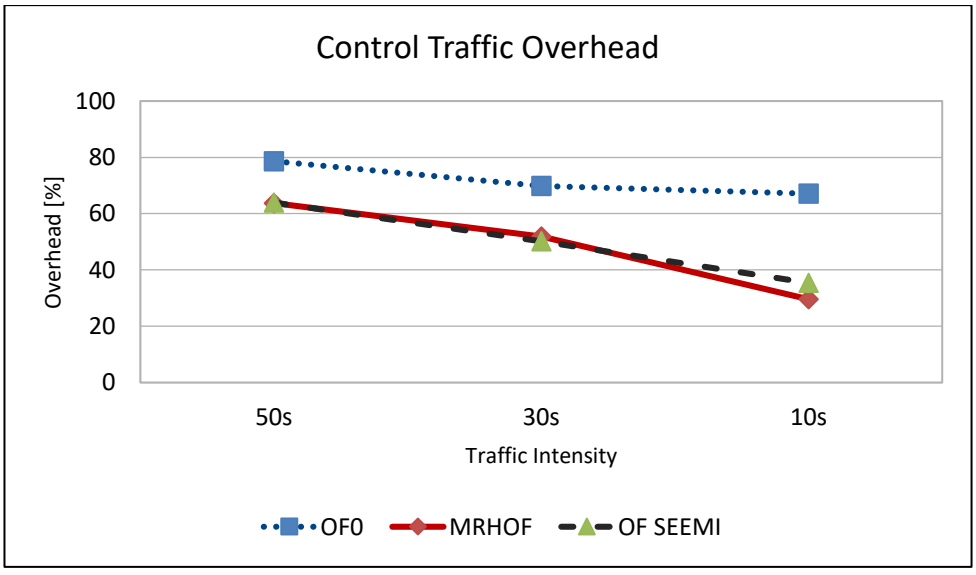

Figure 17: Control Traffic Overhead with 20 nodes.

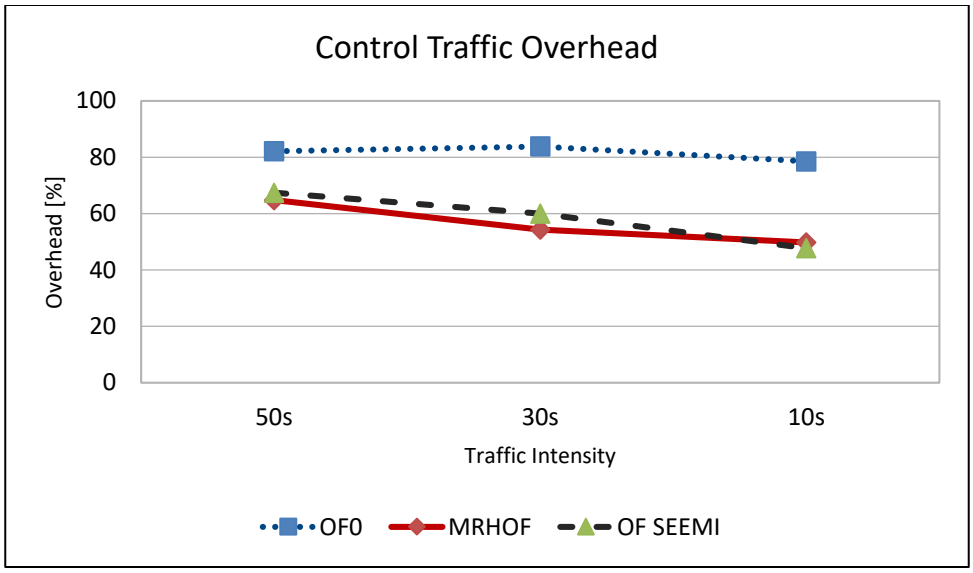

Figure 18: Control Traffic Overhead with 30 nodes.

Figure 19, 20 and 21 are showing the results of throughput derived from a loss function in percentage. Figure 19 is showing that all objective functions are giving throughput above 90% but the OF Seemi is giving 99 to 100% throughput which is higher than both the other two OFs in less dense network with all scenarios of high intensity traffic.

Figure 20 and 21 is showing that the performance of OF0 is decreasing much faster as we are moving towards more dense networks with higher traffic intensities. While the OF Seemi and MRHOF is giving same performance in cumulative perspective and still giving 90% above throughput in more dense networks even with higher traffic intensities.

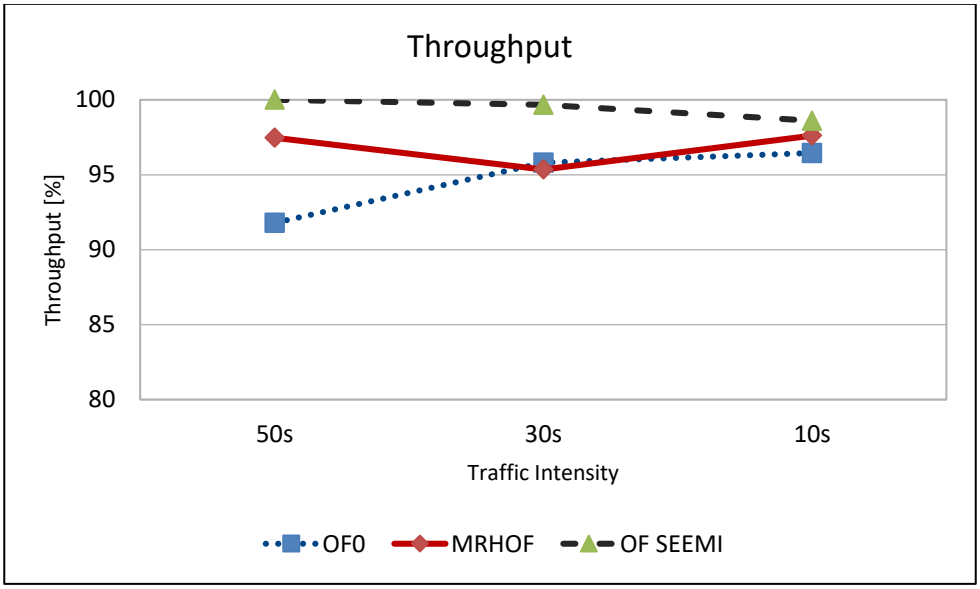

Figure 19: Throughput with 10 nodes.

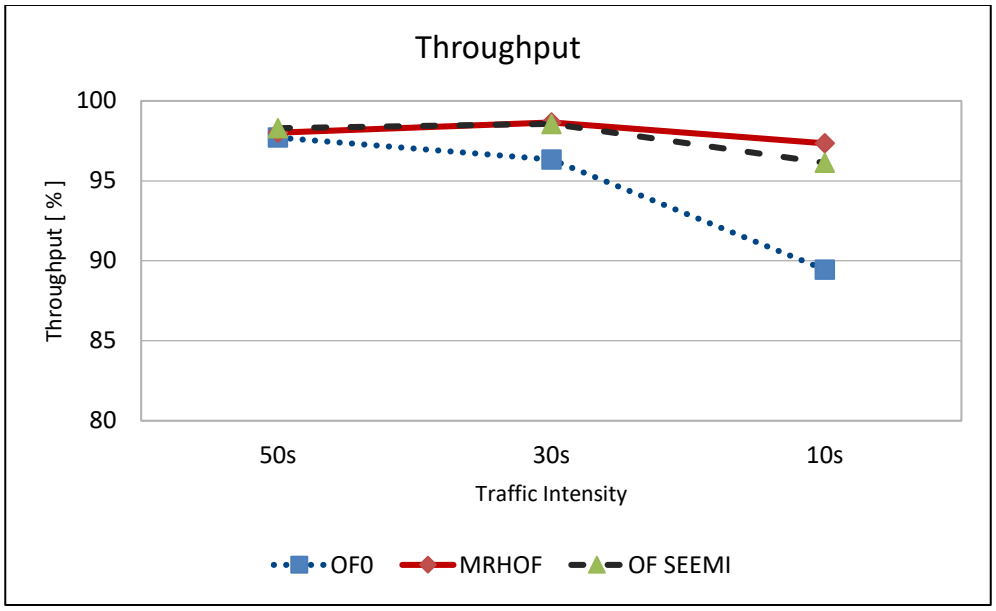

Figure 20: Throughput with 20 nodes.

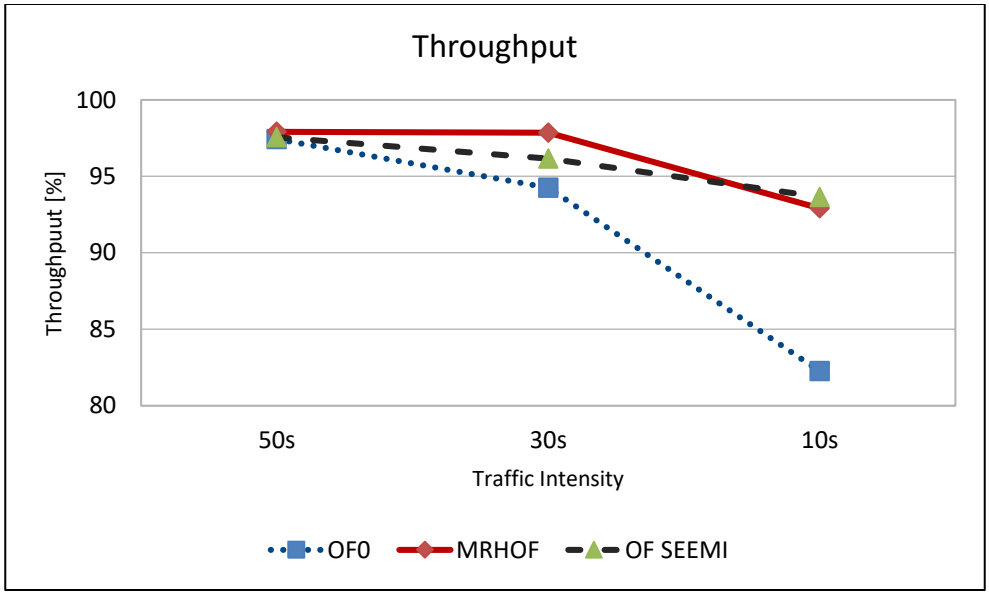

Figure 21: Throughput with 30 nodes.

4.2. Low Traffic Intensity

In low traffic intensity, we have taken time intervals of 05, 10 and 15 minutes in highly dense network having 40, 60 and 80 number of motes to give a comparative analysis for non-real time traffic scenarios. Our results are showing that MRHOF and OF Seemi is performing same still better than OF0 and giving less overhead in the network of 40 motes as shown in figure 22.

Figure 23 and 24 and showing that MRHOF and OF Seemi is giving same control traffic overhead even in more dense networks with 60 and 80 motes. These figures are also showing that

OF0 is giving poor performance as we are going towards denser network with 60 and 80 motes. But still all objective functions are giving overhead above 90% which means almost all resources of a mote are being consumed just for the control traffic.

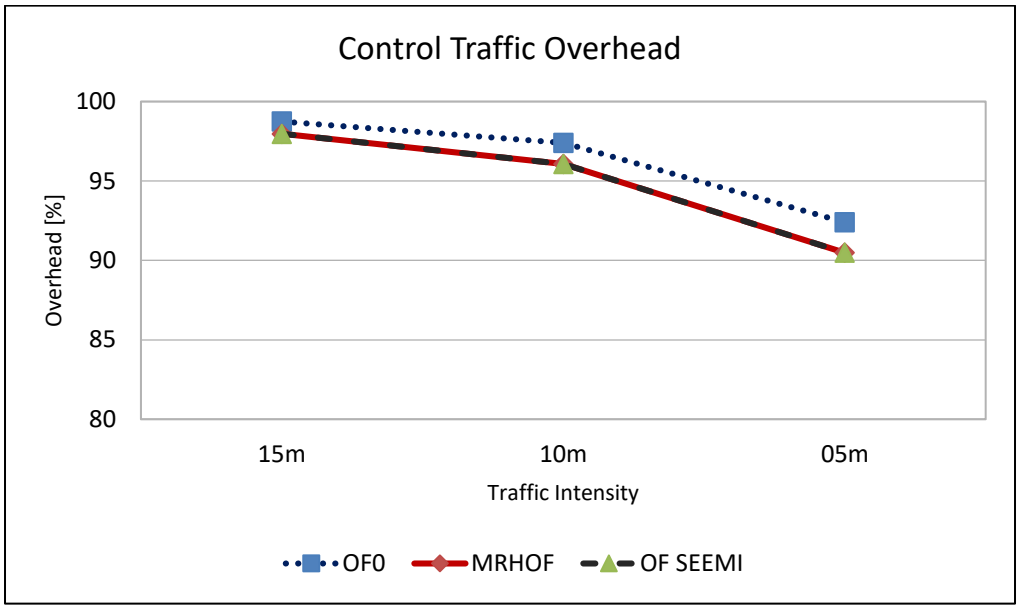

Figure 22: Control Traffic Overhead with 40 nodes.

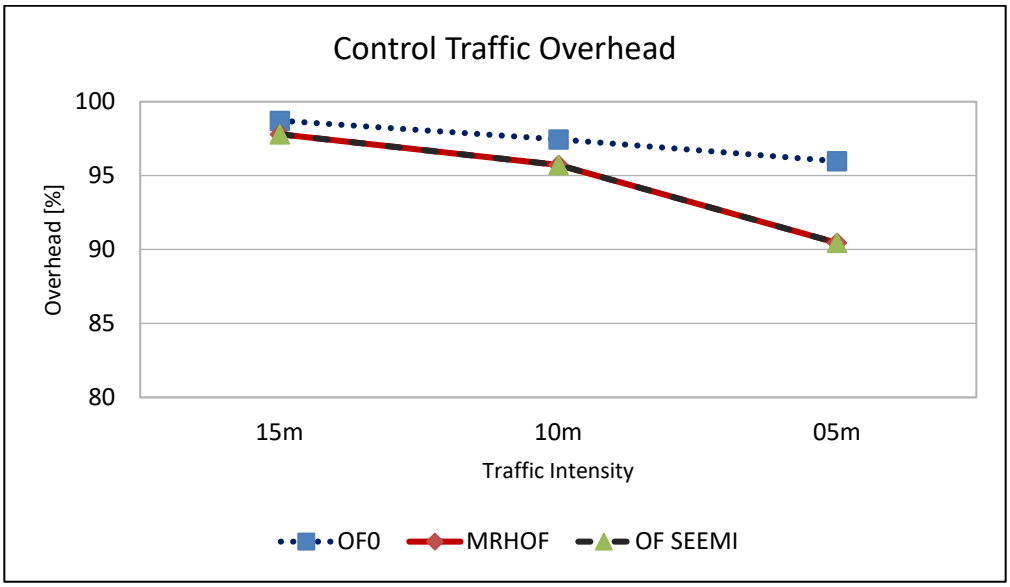

Figure 23: Control Traffic Overhead with 60 nodes.

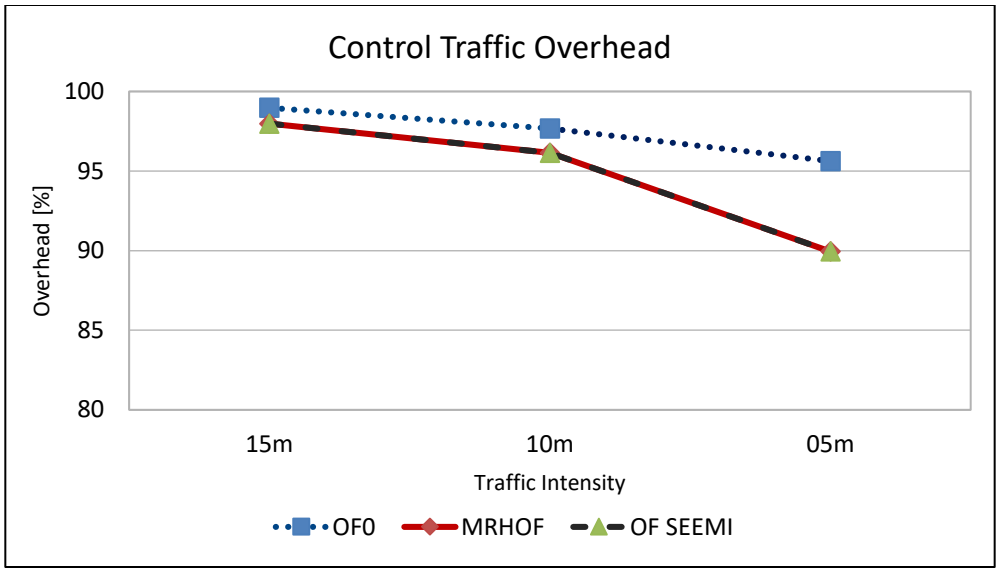

Figure 24: Control Traffic Overhead with 80 nodes.

Results in figure 25, 26 and 27 are showing the throughput for low intensity traffic in highly dense networks. We can see that throughput is more than 95 % in all scenarios. Figure 25 is showing that all objective functions are giving 100% throughput with in a dense network of 40 motes with low traffic intensity.

Figure 26 and 27 are depicting the fact that even OF0 is giving throughput above 97 % in highly dense network. While the performance of OF Seemi and MRHOF is same and in between 99-100 %.

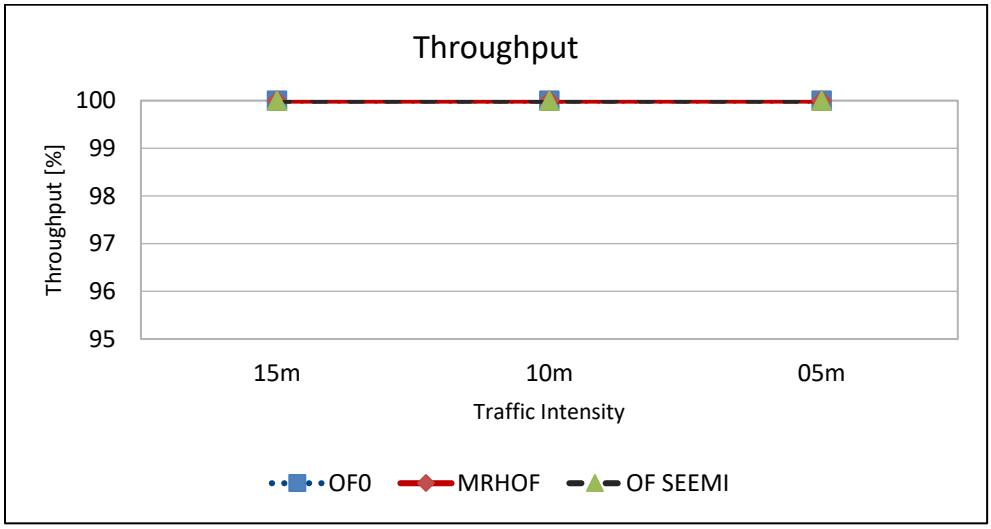

Figure 25: Throughput with 40 nodes.

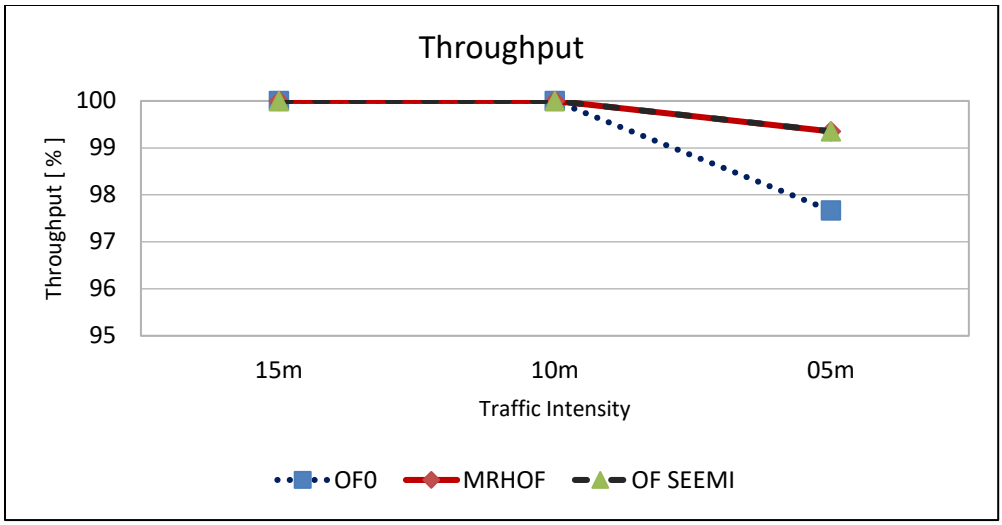

Figure 26: Throughput with 60 nodes.

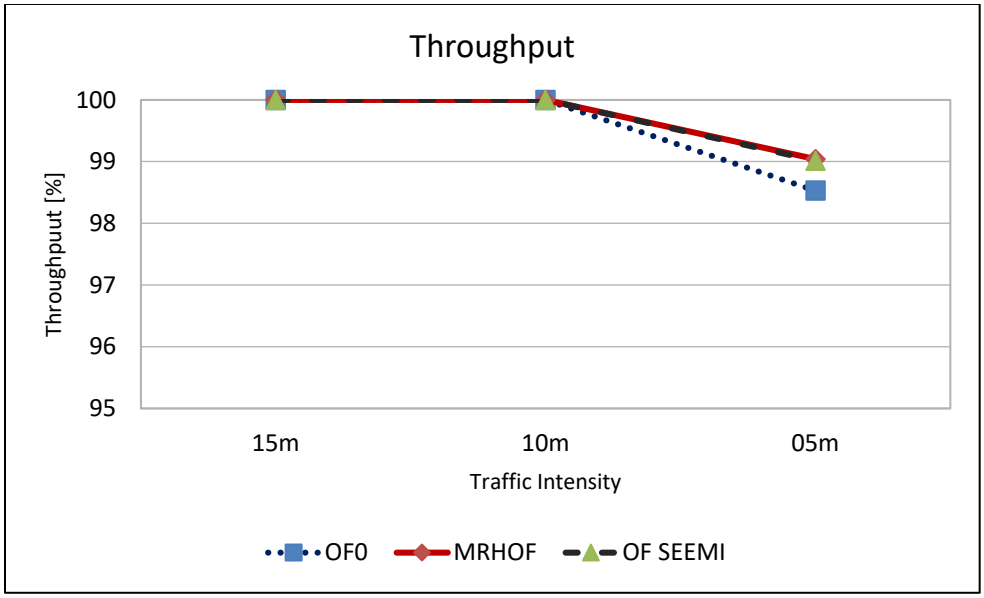

Figure 27: Throughput with 80 nodes.

Conclusion & Future work

From the results, we can conclude that control traffic overhead is in between 20-60% during high traffic intensity of MRHOF and OF Seemi. While the overhead of OF0 in much more than the others. On the other hand, in the case of low traffic intensity all objective functions are giving more than 90% overhead. On the bases of our observations, we will suggest that there is a need to introduce long periodic hoping instead of sending DIO after short time using trickle timers, once a DODAG is being created in the RPL network. Our observations of throughput for low intensity network suggests that any of the objective function can be used for non-real time application. While in the case of high intensity traffic OF0 should be avoided because the throughput of OF0 decreased in denser networks.

For the future work, we will evaluate our objective function along with mobile nodes. We will also try to evaluate our results for reliable application using TCP at transport layer. In the broad perspective we can see that our objective function can further be improved by introducing the hysteresis part. To find the best threshold for hysteresis we have to simulate our results along with different threshold values and compare them in terms of traffic overhead. Our proposed objective function can further be improved with the help of optimal weights. We will use some machine learning techniques to find out the optimal weights for the parameters of our proposed objective function.

References

[1]. Madakam, S., Ramaswamy, R., & Tripathi, S. (2015) Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications, 3*, 164-173.

[2]. Atzori, L., Iera, A., & Morabito, G. (2017). Understanding the Internet of Things: Definition, potentials, and societal role of a fast-evolving paradigm. *Ad Hoc Networks,56*, 122-140.

[3]. Marques, G., Garcia, N., & Pombo, N. (2017). A Survey on IoT: Architectures, Elements, Applications, QoS, Platforms and Security Concepts. In *Advances in Mobile Cloud Computing and Big Data in the 5G Era* (Vol. 22, pp. 115-130). Springer, Cham.

[4]. Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020. (2013, December 12). Retrieved from https://www.gartner.com/newsroom/id/2636073

[5]. Yassein, M. B., Shatnawi, M. Q., & Al-zoubi, D. (2016). Application layer protocols for the Internet of Things: A survey. In *2016 International Conference on Engineering & MIS (ICEMIS)*. Agadir, Morocco: IEEE.

[6]. Shelby, Z., et al. (2014, June). RFC 7252 - The Constrained Application Protocol (CoAP). Internet Engineering Task Force (IETF).

[7]. Accettura, N., Grieco, L. A., Boggia, G., & Camarda, P. (2011). Performance Analysis of the RPL Routing Protocol. In *2011 IEEE International Conference on Mechatronics*. Istanbul, Turkey: IEEE.

[8]. Farrel, A., et al. (2012). RFC 6550- RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. Internet Engineering Task Force (IETF).

[9]. Qasem, et al. (2017). Load Balancing OF for RPL. Internet Draft. Internet Engineering Task Force (IETF).

[10]. Ji, et al. (2017). Traffic-aware OF. Internet draft. Internet Engineering Task Force (IETF).

[11]. Gnawali, O., & Levis, P. (2012). RFC 6719 – The Minimum Rank with Hysteresis Objective Function (MRHOF). Internet Engineering Task Force (IETF).

[12]. Alvi, A. S., Hassan, H., & Mian, N. A. (2017). On the energy efficiency and stability of RPL routing protocol. *In 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC),* Valencia, Spain: IEEE.

[13]. Mulligan, G., et al. (2007). The 6LoWPAN architecture. *In EmNets '07 Proceedings of the 4th workshop on Embedded networked sensors,* Cork, Ireland: ACM New York, NY, USA.

[14]. Thubert, P. (2012). RFC 6552 - RPL Objective Function Zero. Internet Engineering Task Force (IETF).

[15]. Zheng, T., Ayadi, A., & Jiang, X. (2011). TCP over 6LoWPAN for Industrial Applications: An Experimental Study. *In 2011 4th IFIP International Conference on New Technologies, Mobility and Security*, Paris, France: IEEE.

[16]. Ayadi, A., et al. (2010). TCP header compression for 6LoWPAN. Internet draft. Internet Engineering Task Force (IETF).
[17]. Waitzman, D., et al. (1988). RFC 1075 - Distance Vector Multicast Routing Protocol. Internet Engineering Task Force (IETF).
[18]. Mannan, P., & Jayavignesh, T. (2016). Alternate simplistic approach to solve count-to-infinity problem by introducing a new flag in the routing table. *In 2016 Second International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN),* Kolkata, India: IEEE.
[19]. Gomez, C., et al. (2018). TCP Usage Guidance in the Internet of Things (IoT). Internet Draft. Internet Engineering Task Force (IETF).
[20]. Ghaleb, Al-Dubai, A., Ekonomou, E. (2015). E-Trickle: Enhanced Trickle Algorithm for Low-Power and Lossy Networks. *In 14th IEEE International Conference on Ubiquitous Computing and Communications (IUCC-2015),* Liverpool, UK: IEEE.
[21]. Djamaa, B., Richardson, M. (2015). Optimizing the Trickle Algorithm. *IEEE Communications Letters*, *19*, 819-822.
[22]. Yassein, B. M., Aljawarneh, S., Ghaleb, B., Masa'deh, E., & Masa'deh, R. (2016). A New Dynamic Trickle Algorithm for Low Power and Lossy Networks. In 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, Morocco: IEEE.
[23]. Levis, P., Clausen, T., Hui, J., Gnawali, O., Ko, J. (2011). RFC 6206 - The Trickle Algorithm. Internet Engineering Task Force (IETF).
[24]. Contiki: The Open Source OS for the Internet of Things. Retrieved from http://www.contiki-os.org/
[25]. Vasseur, J., et al. (2012). RFC 6552- Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks. Internet Engineering Task Force (IETF).
[26]. The official git repository for Contiki, the open source OS for the Internet of Things. (2011). Retrieved from https://github.com/contiki-os/contiki