Load Balancing in RPL Protocol

Submitted By

Bhavik Patel

15MCEN16

DEPARTMENT OF INFORMATION TECHNOLOGY

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Load Balancing in RPL Protocol

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (Network Technologies)

Submitted By

Bhavik Patel

(15MCEN16)

Guided By

Dr. Vijay Ukani

Dr. Gaurang Raval

DEPARTMENT OF INFORMATION TECHNOLOGY

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Certificate

This is to certify that the major project entitled **"Load Balancing In RPL Protocol"** submitted by **Bhavik Patel (15MCEN16)**, towards the partial fulfillment of the requirements for the award of degree of Master Of Technology of Computer Science and Engineering (Network Technologies) of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-II, To the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Gaurang Raval
Guide & Associate Professor,
Coordinator M.Tech - CSE(N.T),
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Vijay Ukani
Guide & Associate Professor,
CE - Department
Institute of Technology,
Nirma University, Ahmedabad

Dr. Madhuri Bhavsar
Professor and Head,
IT Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Bhavik Patel**, Roll. No. **15MCEN16**, give undertaking that the Major Project entitled "**Load Balancing In RPL Protocol**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (Network Technologies)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place:

Endorsed by

Dr. Gaurang Raval

Dr. Vijay Ukani

(Signature of Guide)

Acknowledgements

Studying routing issues in IoT was exciting part of my learning experience. The presentation of this report gives me an immense pleasure and a sense of satisfaction. No system is created entirely by an individual. Many people have helped to create this system and each of their contribution has been valuable. Proper organization of concept and analysis of the system is due to keen interest and helping hand of my teachers and colleagues. It is joyous to express my gratitude and respect to all those who inspired and helped me in completion of this project. I take this opportunity to acknowledge their support to me.

First of all I would like to extend my sincere thanks to my Parents and God, for their support and blessings.

My sincerest gratitude to my guide, Dr. Vijay Ukani and Dr. Gaurang Raval for molding my thoughts and vision towards this area. I am very much thankful to Dr. Madhuri Bhavsar, Head of Department, who was a constant source of inspiration.

In my daily work I have been blessed with a friendly and cheerful group of fellow students. They have always been encouraging and motivating me throughout the Project.

- Bhavik Patel

15MCEN16

Abstract

The IETF Routing Over Low-power and Lossy Networks working group has newly deploy IPv6 Routing protocol for Low-Power and Lossy Networks, i.e, the RPL protocol. The RPL routing protocol is built for cope the typical need for wireless sensor networks. Like, efficient and reliable data collection in Low-power and Lossy networks. Specifically, it generates a Destination Oriented Directed Acyclic Graph (DODAG) for data dissemination. However, owing to the random deployment of sensor nodes in vast areas, and the different traffic patterns in the network, some sensor nodes may have much more burden in terms of packets forwarded than others. Such unbalanced burden distribution will result in these sensor nodes quickly exhausting their energy, and therefore network lifetime decrease. By using the combination of RED and M-RPL we can achieve the balance on a network, By using RED, detect the congestion on a network. And using M-RPL share the burden from one node to another node by allocating new path. As a result increased the network lifetime and packet delivery ratio.

Abbreviations

6LOWPAN	IPV6 Low Power Personal Network
AODV	Ad hoc On-Demand Distance Vector
ARED	Adaptive Random Early Detection
COAP	Constrained Application Protocol
DAG	Directed Acyclic Graph
DAO	Destination Advertisement object
DIO	DODAG Information Object
DIS	DODAG Information Solicitation
DODAG	Destination Orinted DAG
D-RPL	Dynamic RPL
ETX	Excepted Transmission Count
IETF	Internet Engineering Task Force
IPSO	Internet Protocol for Smart Objects
IoT	Internet Of Things
LLN	Low-Power and Lossy Networks
MAXth	Maximum Thresold
MAXp	Maximum Packet
MINth	Minimum Thresold
M-PATH	Multi-Path
OF	Object Function
OLSR	Optimal link State Routing Protocol
QU-RPL	Queue-Utilization RPL
RED	Random Early Discard
REM	Random Exponential Marking
RPL	Routing Protocol for Low-Power and Lossy Networks

Abbreviations

RRED	Robust Random Early Discard
RED-PD	Random Early Discard with Preferential Dropping
SFB	Stochastic Fair Blue
TCP	Transmission Control Protocol
WSN	Wireless Sensor Network
Wq	Weighted Queue

Contents

List of Figures

Chapter 1

Introduction

Internet of Things (referred as IoT further in the report) a termed stamped by British Organizer Kevin Ashton in 1999 referring to Auto Id objects communicating through Radio-Frequency Identification (RFID). IoT is basically a network of things i.e machines connected together and communicating. The main idea behind IoT is that every physical thing with embedded circuitry can be connected to the internet. It revolutionizes the idea that everything can be used to collect data and map information. Basically, IoT is an extension of sensor ad-hoc networks provided with additional connectivity capability. Speculations are that there would be more than 50 billion objects in IoT by 2020.

Moving towards a more sophisticated definition of IoT as defined by CASAGRASA[1] worldwide system framework, connecting physical and virtual protests through the misuse of information catch and correspondence abilities. This framework incorporates pre-work and advancing Internet and system improvements. It will offer particular verification of object, sensor and joint ability as the reason for the deployment of independent cooperative services and purposes. These can be categories by a peak degree of automatic data acquisition, transfer event, connectivity of network and uncooperative. The things in IoT are defined in[2] a thing, in the IoT, can be a person with a heart screen embed, a animal with a biochip transponder, a vehicle that has strived in sensors to alarm the operator when tire density is low - or

any other regular or natural protest that can be assign an IP address and outfitted with the ability to exchange information over a network. The applications of IoT in various different sectors are depicted in the following Figure. When gazillion number

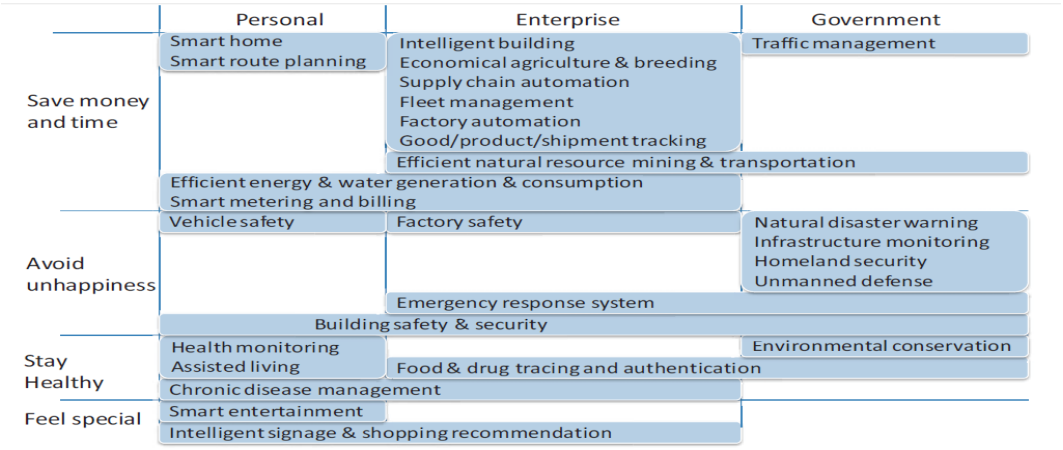

Figure 1.1: Applications of IoT [3]

of devices is connected together and communicating there are many issues arising in transfer and reception of the data as mentioned in [3]. In this study, the major challenges related to IoT are addressed and followed by routing issues in IoT.

1.1 Challenges in IoT

The following sections address the basic challenges in designing IoT as mentioned in [4]. The following Figure 1.2 gives brief information about some of the challenges in IoT.

1.1.1 Battery-Constrained Sensors

The major challenge is the things in IoT are sensors or actuators having small memory space and low-power. These sensors are constrained as there are thousands of sensors in a network and providing high power to them could result in major power usage. So the major challenge is designing sensors which make use of less power.

One of the ways to solve this opportunity is that highly accurate sensors consume

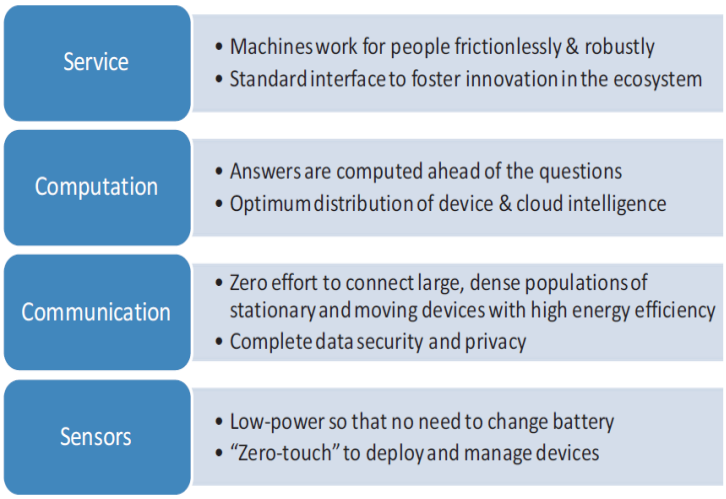

Figure 1.2: Challenges of IoT [3]

more power so make use of the array of less accurate sensors. Another way is that design low power encoding algorithm as encoders consume more power than decoders. So a solution is moving complexity to decoders rather than encoders.

Another important aspect is digital circuits consume less power as compared to analog circuits so it's advisable to make use of digital circuits. Finally, if none of the above solutions works we can use some energy harvesting mechanisms that use solar, wind, hydro energy or vibrations to provide power to sensors.

1.1.2 Naming and Identity management

The IoT will connect billions of devices or objects and each should be uniquely identifiable in the network. Each object sensor has one of a kind character over the web. The effective naming and identity management framework required that can dynamically assign and manage uncommon identity for such a vast number of objects.

1.1.3 Connectivity

The major challenge is providing connectivity between various things in IoT to exchange information. In the network of thousands of sensors, it is very difficult to

provide direct communication between sensors and sink so, it is advisable to provide in-network processing. The major issue in providing in-network processing is the redundancy of information. Proper algorithms need to be formulated.

The issue is sensors can be mobile for example, in vehicle-vehicle communication. It is very important to study various communication protocols that provide mobile communication and are less power consuming as mentioned in [5]. Moreover, in wireless communication lot of bandwidth is wasted. To save the signaling overhead it is necessary to propose better hybrid and cooperating communication protocols.

1.1.4 Data confidentiality and encryption

Sensors will sense the data and send it to network so confidentiality of information is very important where we can use some standard encryption/decryption to encrypt the data and send over the network and another device can decrypt it.

1.1.5 Self-Management and Analysis

Analysis of data provided by humans is difficult so here we are talking about sensor data provided through machine-machine communication. The proper analysis of data obtained from these sources is highly required to make complex decisions. Moreover, it is necessary that only the useful data is collected from the sensors and not all the data. So, here there is a need for context-aware algorithms.

The IoT is used in various critical sectors including health care and military. In these sectors, the security and privacy of data are very important. The security of data requires many complex algorithms to be designed and employed in the IoT. Many cryptographic algorithms and firewalls are used to provide security and privacy to data.

Another important issue is the management of things without using any human intervention. The human intervention is not even possible as thousands of sensors are lying in the network and these sensors require automated testing and maintenance algorithms. The automated maintenance procedures would make sure that

the sensors are well maintained at regular intervals and are working during their duty-cycles.

1.1.6 Smart Service

The IoT is basically designed to provide smarter service to people and reduce their manual work. Such smarter service can be provided via employing artificial intelligence and smart sensing algorithms. The usage of these algorithms requires more power consumption.

Another important aspect is standardization of protocols. To provide reliable and smarter service to people it is important to standardize the protocols employed because if different protocols are employed in things of the same network than they would exhibit different behaviors. Hence, the use of standard protocols is necessary.

1.2 Routing issues in IoT

Routing is the process of finding optimal p_a of information from one node to another. The concern here is how routing is done in IoT. To address the concern the IPSO alliance has proposed RPL routing protocol in [6] based on routing issues mentioned below.

1.2.1 Power Consumption

As discussed in the above section's power is an important constraint in the design of IoT. Moreover, this constraint also hinders the routing process of IoT. So, to address this issue the routing protocol should be such that it consumes less power and yields optimal results.

The communication medium used by the IoT for communication also is one issue why routing over such medium incurs power consumption. The communication medium is such that it makes use of already utilized communication lines. Moreover, the memory footprint of such devices is required to be small so routing table storage could lead to an issue.

1.2.2 Lossy Networks

As mentioned earlier the IoT makes use of already utilized power line communication which is typically used by landline communication. Now utilizing such a communication medium could lead to more number of packet losses and thereby raising a requirement for making efficient error control routing protocol.

In a case of IoT networks, it is based on the concept of under-utilization as they are power constrained. The network rather finding alternative paths and resending the lost packets makes use of full convergence procedure hence repeating the routing process again.

1.2.3 Auto-Configuration and Management

The self-management issue in IoT is one of the key concerns. To solve this issue it is necessary that routing protocol provides the functionality for auto-configuration. The IPSO has provided a way to solve this issue.

The approach is to make use of IPv6 header. The IPv6 has an inbuilt header along with a lengthy addressing scheme.

1.2.4 Link Quality

The link quality should be really better in traditional networks as they carry voice and video traffic but when we are considering lossy networks they do not transmit such heavy traffic. The link quality can be compromised in lossy networks.

Hence, instead of full global convergence, the use of finding an alternate path is more viable.

1.3 Objective of Study

The use of IoT is progressing day-by-day and routing is the key area to make IoT possible. The objective of this study is basically analyzing the routing protocol (RPL) of IoT. In event-driven network traffic generates simultaneously. Due to generation of this type of traffic congestion occurred on the getaway node. Owing

to the congestion data packet did not reach the destination. As a result packet loss. To resolve this type of problem we use RED and M-RPL technique. By using this technique optimize the network by sharing the gateway burden to other node and providing a different path to reach the destination.

1.4 Thesis Structure

The narrated work concerning the issue is outlined in chapter 2 and in chapter 3 we exhibit our strategy we embraced for this research. The implementation and in chapter 4. Finally, chapter 5 shows the result and analysis of implementation, in chapter 6 we exhibit the outcome of the research and planned investigation work created of this review.

Chapter 2

Literature Survey

2.1 RPL

The IETF formed a group call ROLL to find a routing protocol standard for IoT in 2008. The ROLL group came up with an algorithm called Routing Protocol for LLN (Low Lossy Network) based on various analysis made on low-power lossy networks. This proposed protocol was employed on IP layer instead of link layers like other routing protocols.

2.1.1 RPL Terminology

- DAG: All the ends are connected in such a style that no circles exist in a network topology.
- DAG Root: As a result the graph is acyclic; all DAGs should have a minimum of one this origin and every one way eliminate at a DAG origin.
- Destination-Oriented DAG (DODAG): A DAG rooted at one target, i.e., at one DAG root (the DODAG root) with no friendly ends.
- DODAG root: In this origin might act as an edge router for the DODAG; especially, it's going to mixture routes within the DODAG and should spread DODAG routes into different routing.

- Virtual DODAG root: The results of 2 or additional RPL routers, for example, 6LoWPAN Edge Routers, coordinative to synchronize DODAG state and act jointly as if they're one DODAG root, with relation to the LLN.
- Up: Up refers to the direction from leaf nodes towards DODAG roots.
- Down: Down refers to the direction from DODAG roots towards leaf nodes.
- Rank: A nodes rank characterizes the nodes specific position respect to option nodes with relevancy a DODAG root. Rank entirely increases within the downward way and entirely decreases in the Up way. The exact approach Rank is registered relies on the DAGs Objective operate (OF).
- Objective function (OF): An OF is utilized to compute the Rank for optimization path.
- RPL InstanceID: An RPL InstanceID is an uncommon identifier inside a system. DODAGs with the same RPL InstanceID have a similar Objective Function.
- RPL Instance: It is one or greater than one or more DODAGs that belongs to RPLInstanceID.
- DODAGID: This is used to distinguish of a DODAG root. And it is uncommon.
- DODAG Version: This denotes is a specific version of a DODAG with a given DODAGID.
- DODAG Version Number: It is a subsequent counter that is addition by the origin to form a new version of a DODAG.
- Grounded: A DODAG is grounded when the DODAG origin can satisfy the Goal.

- Floating: A floating DODAG isn't considered to own the properties needed to accomplish the goal.

2.1.2 Traffic Flows Supported by RPL

There are three natures of traffic carried by RPL.

- **Multipoint-To-Point (MP2P):** Multipoint-to-point (MP2P) is prevailing movements stream in a few that have some application significance, like giving network to greater streams are chosen hubs that have some application significance, like giving connectivity to the bigger web or center private IP network. RPL assists MP2P activity by providing MP2P destinations to be performed via DODAG origins.

- **Point-To-Multipoint (P2MP):** These (P2MP) is a traffic pattern required by numerous LLN applications. RPL supports P2MP activity by utilizing a goal using the component that arrangements down courses toward objectives, and far away from roots. Goal promotions will refresh steering tables because the underlying DODAG topology changes.

- **Point-To-Point (P2P):** A origin should be ready to route packages to a destination. Hubs inside the network can also have routing information to reach destinations. A packet moves towards an origin till it reaches a relative that incorporates a notable way to the destination. As known later during this document, within the very strained case, that common relative is also the DODAG root. In alternative cases, it's going to be a node nearer to each the supply and destination. RPL additionally supports the case wherever a P2P destination is a one-hop neighbor.

2.2 Working of RPL

RPL is basically a steering protocol that makes a DODAG to find paths to the destination in this case root node. It makes use of two things, one is objective

function and another is a collection of metrics or constraints. The objective function can be made on various constraints like ETX (expected transmissions) or hop count.

RPL basically creates graphs and follows it to route packets across and within networks. The main notion is multiple graphs can be deployed on the same network topology i.e. same physical topology with various logical topologies on top of it. The nodes within the network can join one or more graphs but at a given instance they can be part of only a single graph.

2.2.1 DODAG Building

RPL builds the graph and this process begins from the border router and aggregates to other nodes. RPL makes use of three messages such as DAO (DODAG Destination Advertisement Object), DIS (DODAG Information Solicitation) and DIO (DODAG Information Object).

The root hub begins promoting the data about the graph by making use of the DIO message. The hubs which are present in the neighborhood of the root would receive the DIO message and would now process the DIO message and think over whether to join the graph based on certain constraints and metrics. If the hub joins the graph than it has a path directly connecting to the root. The parent node or the node generating the DIO message is called the "root" of the graph. The node counts the run of itself with respect to the root of the graph and thereby makes use of constraints and metrics. If the node is selected as the forwarding node then it further forwards the DIO packet to those nodes which are in its vicinity but not in the vicinity of the root. If the node is not the forwarding node and just the "leaf node" then it would just calculate the rank and doesn't forwards the DIO message. The nodes thereby receiving the DIO messages would compute their rank and the DODAG graph would be constructed gradually. In this manner, a graph is constructed and there exists a path from the node to the root such that every node in the graph can reach the root node in the graph. Here, it can be seen that all nodes have a path towards the root hence it can be said that a multipoint-to-point

model is formed and it is termed as UPWARD routing. All nodes in the graph have the rank associated with them and are closely linked in the graph with the root of the graph. The rank is computed based on the constraints and metrics such as ETX or Hop count. The steps of creation of DODAG building process is modeled in the following figure.

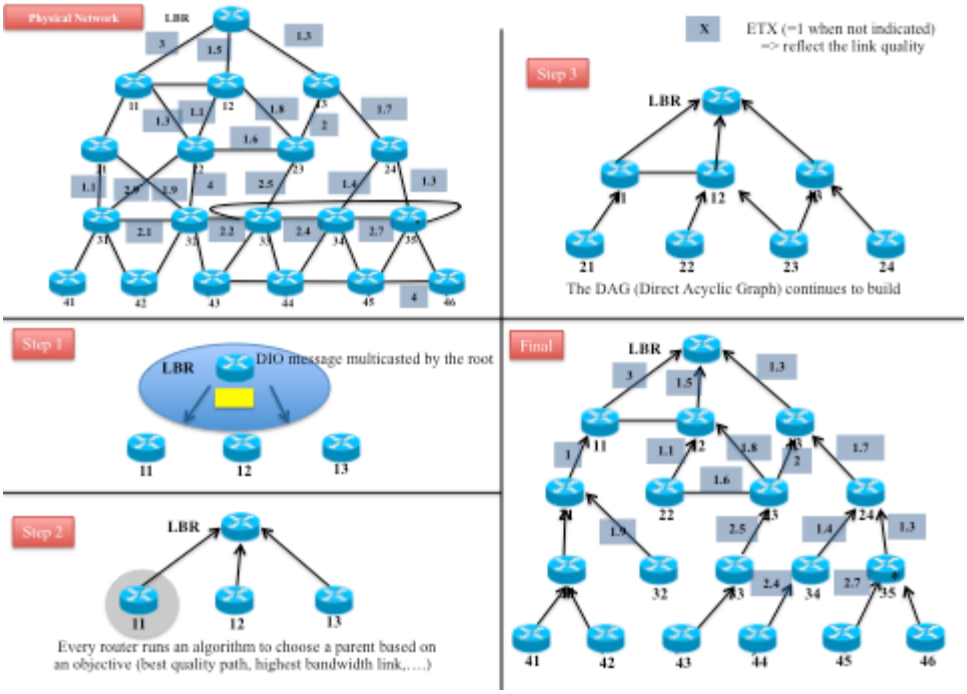

Figure 2.1: DODAG Building [6]

Same as UPWARD routing where the message packets originate from a leaf node to root node there can be a condition when message generates from a root or another network to the leaf node, this requires a path from a root to the leaf node. This kind of routing is termed as DOWNWARD routing. It is obtained by making use of DAO messages. DAO messages are used by the nodes to send routing information to the neighboring nodes or root to store the routing path or not store the path but to keep track from where the messages have arisen. The DAO message is an advertisement message which advertises the various DODAG information to all the nodes in the topology. The prefix reachability matrix is constructed through this

process. The nodes receiving the DAO messages stores the routing information in their respective routing tables. This way of storing information is making use of what is termed as "storing" nodes. The RPL also supports "non-storing" mode. The prefix reachability matrix creates a downward routing path from the leaf nodes to the root node.

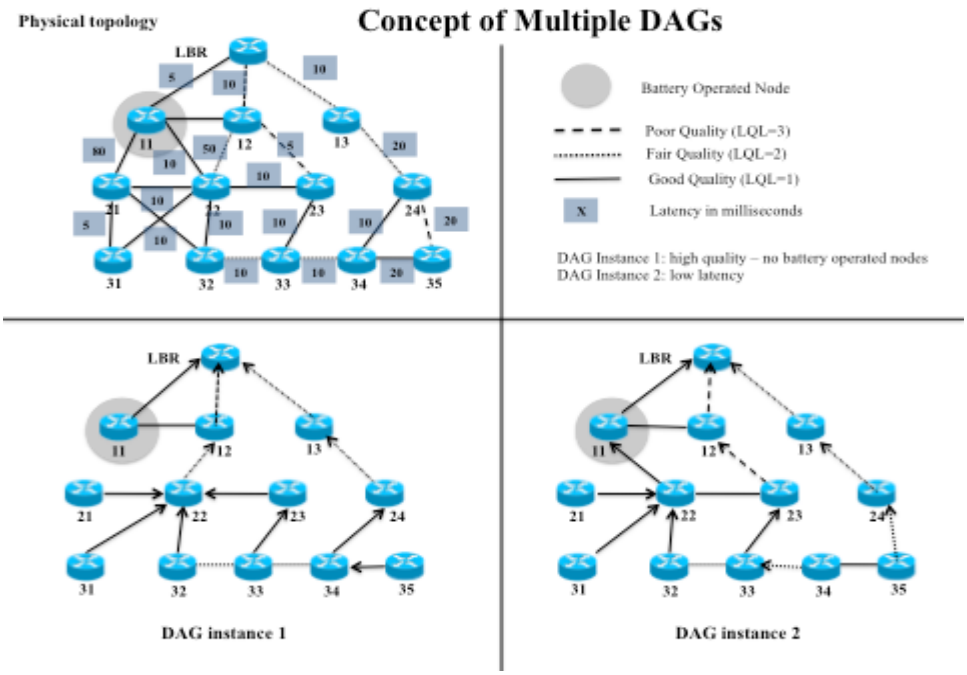

Figure 2.2: Concept of multiple DAGS [6]

2.2.2 Metrics and Constraints

This is used to generate the objective function thereby resulting in a formation of DODAG. The metrics are a scalar quantity which measures the quality of the graph formed or the topology created. Constraints are the limitations that are imposed on the objective function to create the DODAG graph and thereby optimizing the graph building process. These metrics and constraints are something which is not fixed or compulsory but is liable to context and conditions. An example of some metrics is ETX or Hop count and the constraints imposed on them can be that ETX should not be over 55.

2.2.3 Storing and Non-Storing nodes

Storing nodes are the one that would store the routing information in their routing tables implied that they have been provided that routing information is provided to them via the DAO messages. The prefix reachability matrix provides this information. Another method is non-storing where the nodes don't store the routing information but the root node does. Thus the root node has all the routing information. All packets are sent via the root node of the graph. The important aspect here is that there is a tradeoff between both the nodes available in RPL. In storing mode a lot of memory space is to be required with all the nodes which might increase the size of the node. In a case of non-storing mode, there is performance lacking as all packets pass through the root. It should be noted that hybrid mode is not possible.

2.2.4 Loop Avoidance and Detection

RPL provides basically two mechanisms for loop avoidance and detection. Firstly talking the loop avoidance mechanisms, the RPL provides max depth rule in which the rank of the parent cannot be more than the children node and a node cannot go deep in the graph to increase their rank and become greedy to save their power. But it is not always possible to avoid the loops so a loop detection mechanism is required The loop detection mechanism is such that when a packet arises from the root node and is traveling towards the leaf nodes and when any node checks its routing table and finds that the packet needs to go upwards then it is detected that there are some discrepancies involved in the routing of the packet. So the intermediate node doesn't forward the packet further but discards the entire packet without making any amendments in its routing information. Then a local repair is emerged to solve this problem and get correct routing information.

2.2.5 Global and Local repair

Replacement is a key element for every steering convention and alludes to the capacity to repair the steering topology when disappointments happen. When a packet

arises from the root node and is traveling towards the leaf nodes and when any node checks its routing table and finds that the packet needs to go upwards then it is detected that there are some discrepancies involved in the routing of the packet. So the intermediate node doesn't forward the packet further but discards the entire packet without making any amendments in its routing information. Then a local repair is emerged to solve this problem and get correct routing information. As limited repairs happen the diagram may begin to separate from its ideal shape, and soon thereafter it may be important to reconstructing the chart (DODAG) thanks to a corresponding instrument named the "Global Repair". These paper [7] discuss routing protocols proposed by IETF and their issues. Moreover, they include simulation providing proofs for their studies.

2.3 Routing protocol specification

RPL messages are determined as another kind of ICMPv6 control messages. According to[8], the RPL control message divide in two parts first an ICMPv6 header, and second message body. First, an ICMPv6 header which comprises of three fields: Type, Code, and Checksum, and message body including a message base and various choices.

The Type field determines the sort of the ICMPv6 control message tentatively set to 155 if there should arise an occurrence of RPL. The Code field recognizes the kind of RPL control message. Four codes are as of now characterized:

- **DODAG Information Solicitation (DIS) :** This message is identified by the 0*01, and is utilized to request a DIO from an RPL hub. The DIS might be utilized to test neighbor hubs in nearby DODAGs. The present DIS message design contains non-indicated banners and fields.

- **DODAG Information Object (DIO) :** This message is recognized by 0x01

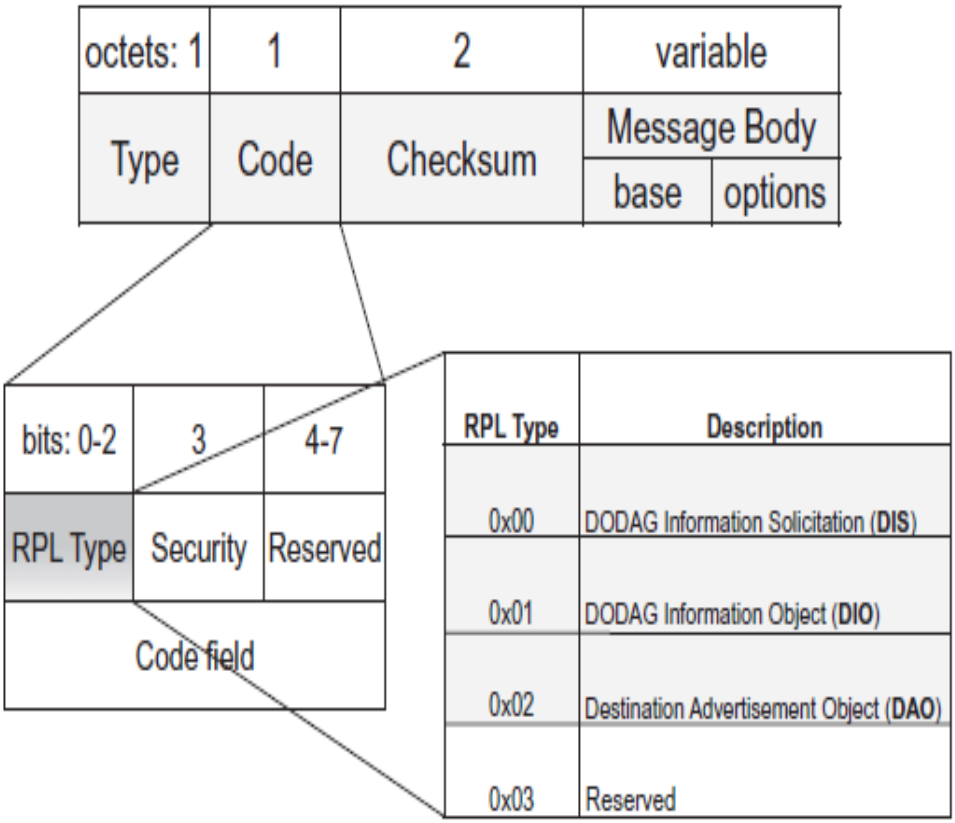

Figure 2.3: RPL Control Message [9]

also it is generated by the DODAG origin to build another DAG and after that sent in a multicast manner by using the DODAG structure. This message conveys important system data that permits a hub to find an RPL occasion, take in its design parameters, pick a DODAG preferred parent set, and keep up the DODAG. The structure of this message is exhibited in Figure 2.4. The primitive DIO Target fields are: (i) RPLInstanceID, It is an 8-bit data started by the DODAG origin that shows the ID of the RPL occurrence in the DODAG, (ii) Version Number, demonstrates the adaptation number of a DODAG that is regularly increased upon each system data refresh, and assists keeping up all hubs synchronized with new refreshes, (iii) Rank, it is a 16-bit field it determines the rank of the hub sending the DIO message, (vi)

Destination Advertisement Trigger Sequence Number (DTSN) is a 8-bit hail that is utilized to keep up descending course.

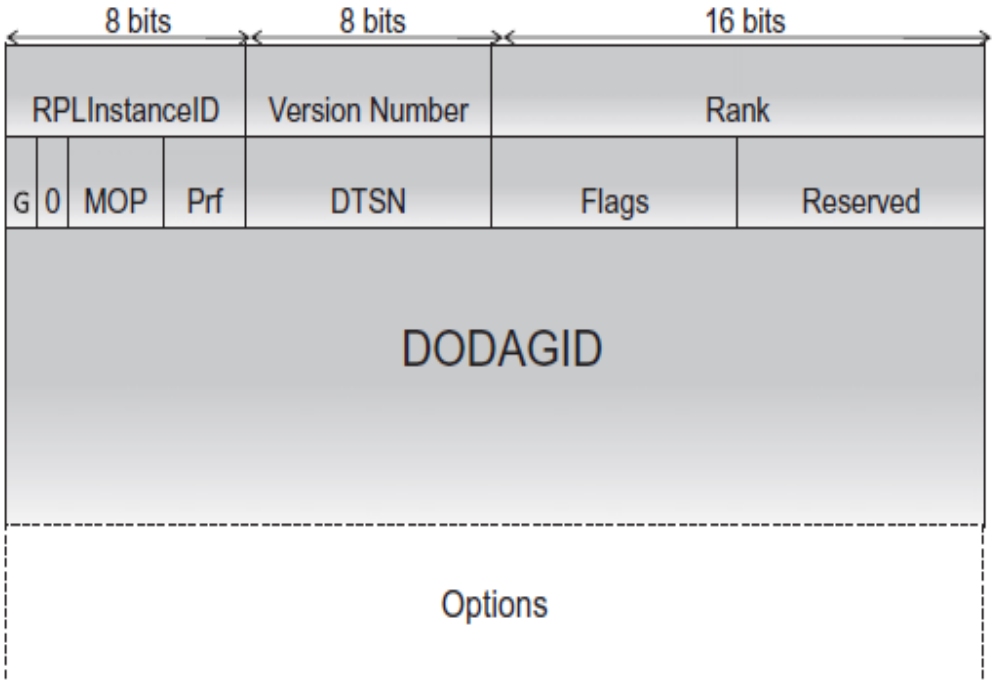

Figure 2.4: DIO Message Structure [9]

- **Destination Advertisement Object (DAO) :** The DAO message is recognized by using these bits 0 * 02 and is utilized to generate switch course data to record the hubs went by along the upward way. Every node sends the DAO message, other than the DODAG origin, to populate the directing tables with names of their kids and to promote their locations and names to their groups. Subsequent to transferring this DAO message through way from a specific hub to the DODAG root through the default DAG highways, a total way between the DODAG root and the hub is built up. Figure 2.5 delineates the organization of the DAO Base Object. As appeared in Figure 2.5, the fundamental this

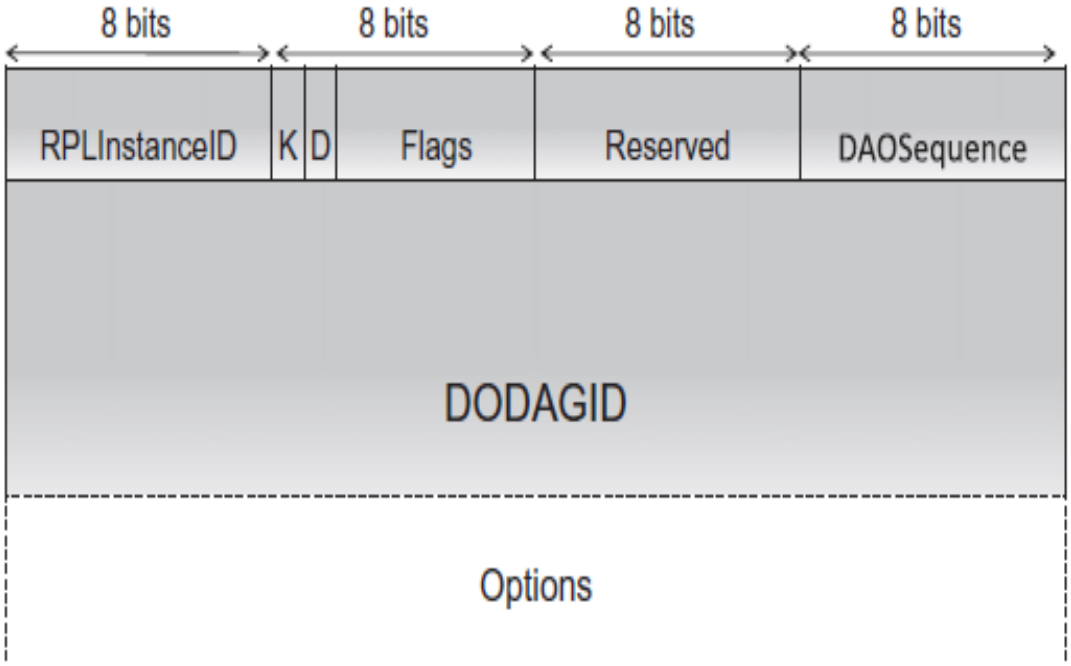

Figure 2.5: DAO Message Structure [9]

message fields are: (i) RPLInstaceID, it is an 8-bit data demonstrates ID of the RPL occurrence as gained of the DIO, (ii) K hail that shows whether and affirmation is required or not in light of a DAO message, (iii) DAO Sequence is an arrangement number increased at each DAO message, (iv) DODAGID is a 128-piece field set by a DODAG root which recognizes a DODAG. This field is available just when hail D is set to 1.

- **Destination Advertisement Object (DAO-ACK) :** The DAOACK message is forwarded as a unicast packet by a DAO beneficiary in light of a unicast DAO message. It conveys data about RPLInstanceID, DAO Sequence, and State, which demonstrate the fruition. The state code is as yet not obviously characterized, but rather code more prominent than 128 mean a dismissal and that a hub ought to choose a substitute parent.

- **Consistency Check :** This message is appropriated to check the secure

message and create protest criticism. A CC message be sent as a secured RPL message [10].

2.4 Work Done In RPL

There are many are many problem are occurred during the generating a topology and transferring the data-packet from one node to another node in RPL protocol. There are some problem are described below.

- Data collection in LLN
- Path Optimization problem
- Transmission delay and Residual enenrgy
- Mobility
- Redundancy
- Load balancing on congested node

2.4.1 Load Balancing

The IETF ROLL social unit has uncovered RFC 6550, which is a steering component named RPL particularly intended for LLNs. RPL pick the steering way by creating a DODAG. Contingent on the precise application, diverse steering measurements will be embraced, as expected transmission number (ETX). Routing path development depending just on one pairwise transmission quality metric won't be prepared to catch the genuine correspondence circumstance. For instance, the pairwise metric may prompt one node with savvy communication quality to each adjacent hub being identified with countless. Amid this case, the parent hub is seriously full and drops an expansive scope of bundles because of cradle confinement.

Along these lines, stack adjustment beneath the non-uniform hub circulation moreover as non-uniform traffic patterns becomes crucial. To alleviate the work

unevenness disadvantage in LLNs, a steering convention should have the ensuing attractive elements.

- Distributed: distributed is tough for any centralize server to get global info regarding energy needed and status of communication of every sensor hub in an extensive scale LLN, a circulated steering convention could be an ought to.
- Non-intrusive: By using the periodic information aggregation and management messages to a get every nodes info in an LLN acquires huge communication overhead and should change normal tasks. Hence, a more robust approach is to notice and signal work imbalance in a non-intrusive manner.
- Reliability: so as to adjust work between sensor hubs, some information traffic is also relayed through a path with faulty transmission link quality. To take care of a routing protocol ought to together think about work equalization and communication link quality

2.4.2 Technique of load balancing algorithm

There are the some technique to solved the load balance on a congested node in topology. That is listed below [11].

- Threshold based approach
- Randomized forwarding mechanism
- Multi-path geographic routing protocol
- LB-RPL protocol
- RED Algorithm (Random Early Detection)

2.4.3 Multi-path geographic routing protocol

RPL is a single path routing protocol do not support the creation of multiple routing tracks between source and destination. M-RPL a multi-path extension of RPL

is proposed that aims to provide temporary multi-path routing during congestion over a path. Multi-path RPL reduces congestion and increases the overall throughput. It is suitable for supporting high data rates as compared to single path RPL. Multipath routing can be used to achieve higher reliability, increased throughput, fault tolerance, Congestion mitigation and whole avoidance. This is because M-RPL initially establishes a single path for data routing as defined by RPL. However, after the detection of congestion at any congested node data splitting start resulting in the creation of multiple routing paths. This is because data is continuously reported and congestion becomes persistent that forces RPL nodes to change their parents. This result in the change of network topology and results in increased delay. Thus, as data is continuously reported the end-to-end delay of M-RPL stabilizes and becomes less than RPL [12]. Multiple paths are created by splitting forwarding rate between both preferred parent (congested node) and alternate parent available in RPL. As a result At the start of data splitting the latency of M-RPL is slightly greater than RPL because of the multiple paths. It is also noticeable that the delay of RPL decreases and becomes similar or less than M-RPL.

2.4.4 QU-RPL : Queue utilization based RPL for load balancing

We point out most of the data-packet dropped under more weight are owing to congestion, and a dangerous burden balancing problem continues in RPL in terms of path root node selection. To mitigate this problem, we use the Queue Utilization QU-RPL Technique based on RPL (QU-RPL). Packet delivery performance is increased examined to the standard RPL in terms end-to-communication. That aims to achieve by allowing each node to select its origin node according to the queue utilization of its intimate nodes as well as it considers their hop distance to the perimeter router [13]. Due to the load balancing capability, QU- RPL is very useful in diminishing the queue losses and improving the packet delivery ratio.

2.4.5 D-RPL : Overcoming Memory limitations

The IPv6 RPL Networks supports both ascending and descending freight. The latter is significant for actuation, for inquiries,and for every bidirectional protocol such as TCP, yet its support is compromised by memory limitation in the nodes. In RPL putting away mode, Hubs store directing sections for every goal in their sub-chart, Constraining the measure of the system and regularly prompting to inaccessible hubs and convention disappointments. We propose here D-RPL, a system that defeats the versatility restriction by retouching putting away mode sending with multi-cast based dispersal. Our change has an insignificant effect on code size and memory utilization[14]. D-RPL is enacted just when memory points of confinement are come to, and influences just the bit of the movement and the portions of the system that have surpassed memory limits.

2.5 Survey Summary

It can be condensed the RPL steering convention distributed in RFC 6550 was intended for productive and dependable information gathering in a low Lossy system. RPL is a capable method, regarding conceding a quick system set-up and limited and correspondence delays, End-End unwavering quality, Energy utilization. To start with it develops a DODAG for information sending. Nonetheless, Due to the uneven organization of sensor hubs in huge regions, The heterogeneous cargo courses of action in the system a few hubs may have the substantially heavier workload as far as bundles sent than others. Its viability can be additionally enhanced as far as overhead, And load adjusting. To moderate the workload awkwardness issue in LLNs, The stack was examined adjusted directing convention in view of the RPL convention to accomplish adjusted workload dispersion among hubs in vast scale LLN. Prior to any hub goes to on disappointment state in straightforward RPL. There are diverse system Used to unraveled the heap adjusting calculation like M-RPL, QU-Utilization, D-RPL procedure to adjust the system cargo over expansive

zone organize however this method tackled the issue bundle are dropped at a clog hub. Be that as it may, an issue is we can't foresee either transmit or dropped. By utilizing this system we increment the bundle dependability in end-to-end hubs and diminish the transmission delay.

Chapter 3

Proposed Work

With the quick development of current web and administration interest for numerous get to has additionally been expanded essentially. In this way, there is immense movement in the network which brings in clog where buffer management acts an essential part. Prior, drop tail and random drop were utilized as buffer management procedures with the TCP. Drop tail experiences tremendous lining delay, lockout, global synchronization issue. To conquer every one of these issues with prior procedures Active Queue Management (AQM) system has been presented. Active queue discipline packets are dropped or set apart before cushion turns out to be full while in prior method does likewise when buffer turns out to be full. We have different algorithm under the AQM as Random early detection(RED), Random Exponential Marking (REM), Blue and Stochastic Fair Blue (SFB), ARED, Re RED, PI controller, Robust random early detection (RRED), RED with Preferential Dropping (RED-PD) etc. But still applying the Active Queue Management Technique some packet is dropped. Owing to the highest threshold value. If the average queue length is higher than the highest threshold then a packet is discarded. As a result, some packet is lost with some probability. Due to the nature of this technique we apply Multi-path RPL (M-RPL) on Active Queue Management Technique. M-RPL protocol allows a node to find alternative paths in case of link failure or congestion occurred on the network. The flowchart in Figure 3.1 shows how it works[15].

3.1 Generation of DODAG

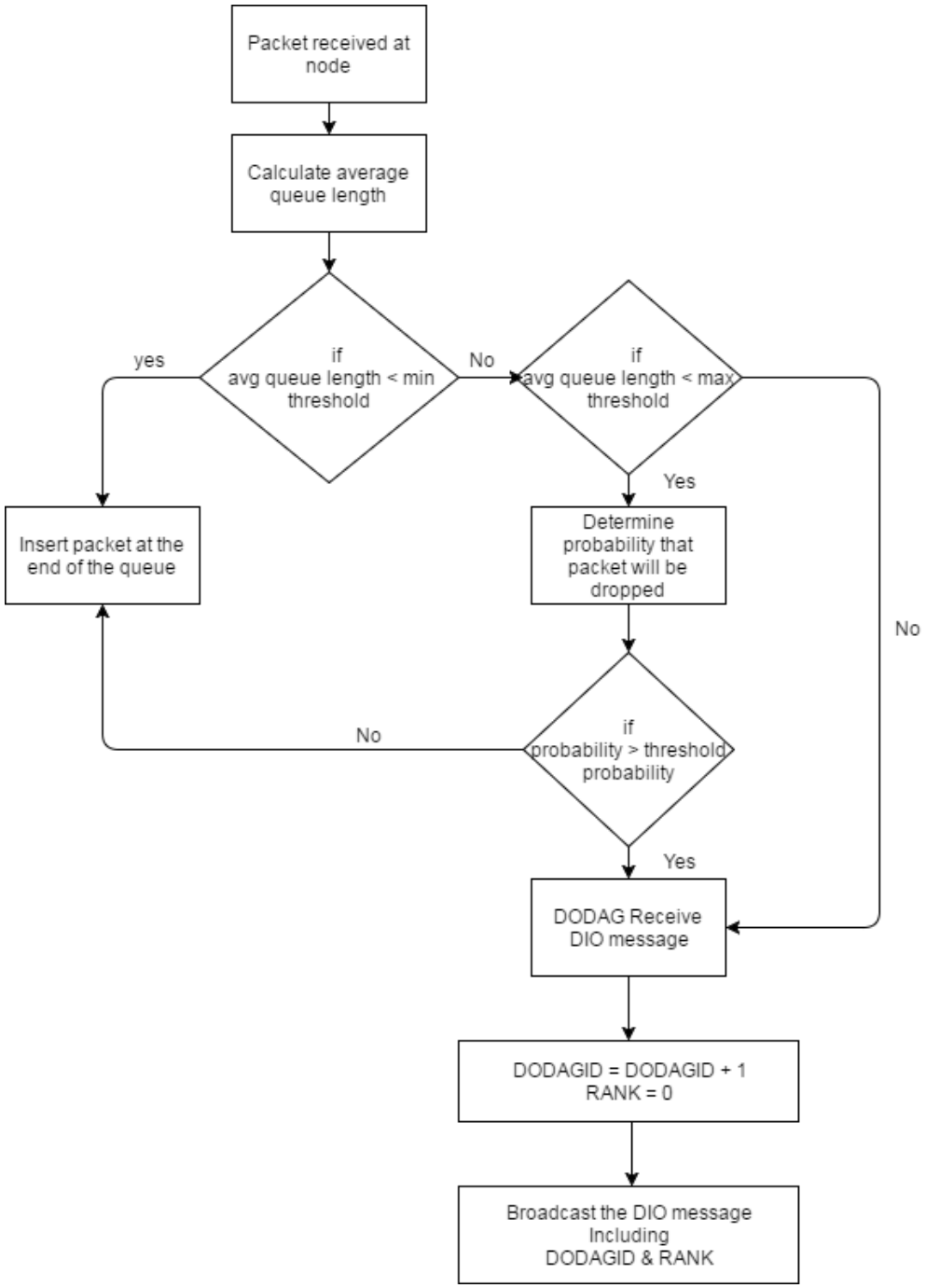

Figure 3.1: Flow of generating a DODAG

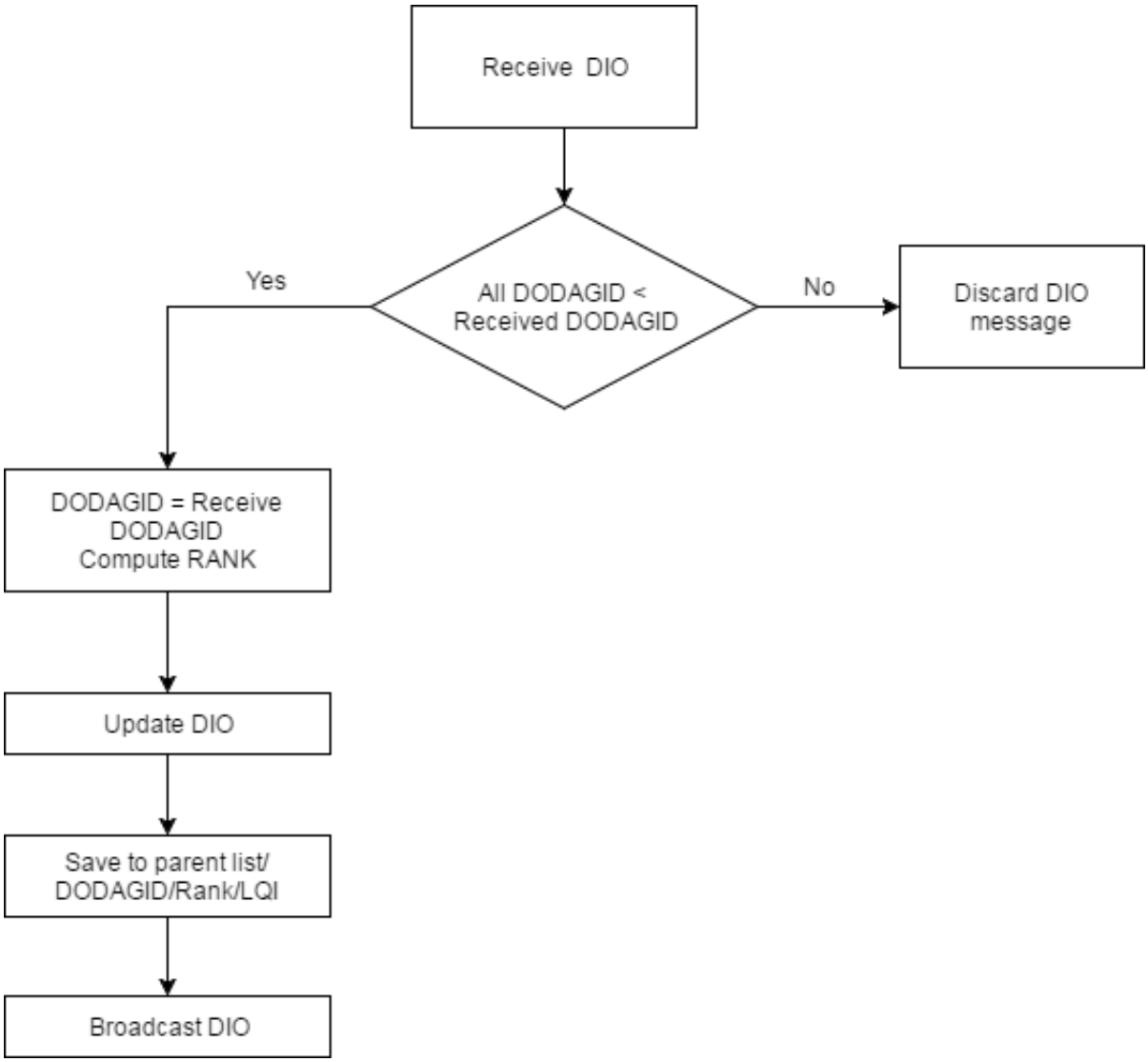

Figure 3.2: Flow of generating a DODAG

In an event-based network particular event occurred simultaneously at that time packet received at a particular node. During this time packet calculate the average queue size of new packet by using formula it is given below.

Formula for average queue length :

$$avg \leftarrow (1 - w_q)avg + w_q q$$

The average queue size is compared the two threshold value one is the smallest

threshold and highest threshold value when the average queue size is smallest threshold value than packet are insert into the queue and no packet is noted. When the average queue size is greater than the highest threshold value then packet sent back to the child node. By using the ICMPv6 control message parent node notifies the child node about the congestion by setting the bit. When the average queue size is inside the lowest and highest threshold value then calculate the probability p_a. Where p_a is the function of average queue size Avg. As Avg varies from min_{th} to max_{th}, the packet-marking probability p_b varies linearly from 0 to max_p: The probability calculated by using formula that describe below.

Formula for calculating probability :

$$p_b \leftarrow max_p(avg - min_{th})/(max_{th} - min_{th})$$

$$p_a \leftarrow p_b/(1 - count.p_b)$$

When the average queue size is greater than the maximum threshold value then packet sent back to the child node and give notification about the choose the different path from parent list. The DODAG generation is mainly based on the neighbor discovery, which serves the two main procedures, (1) broadcasting of a DIS message by a source node to the sink node to request DODAG generation, (2) broadcasting of a DIO message by the sink node to construct a new DODAG. When the root node n receives the DIO message it checks the DODAGID with the DODAGID that is contained in the DIO message. If the DODAGID is larger than all the DODAGIDs of the node N, which shows the message of a new DODAG formation, the node N restores its DODAGID with received DODAGID and specifies its rank that is equal to received rank incremented by rank 1. If received DODAGID is similar to the node Ns DODAGID, the sender can be seen as another alternate parent node (with higher rank) or a sibling node (with equal rank) [12]. Then node N also updates the DIO message. But, if the received rank is more than the node Ns rank, the node N-dump the DIO message since the sender can be its down node in order to

avoid loop creation. After having finished updating the DIO message, the node N creates a new entry in the parent list that involves applicant parents and relative that can be practiced if the currently elected parent loses its routing ability. In the generation manner of a network, the node N saves the sender node ID, DODAGID, rank, and LQI information to the parent list. Then, it broadcasts the updated DIO message. The constructed DODAG expires after a prearranged period of time.

Example

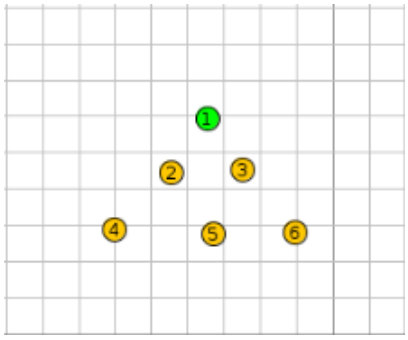

Figure 3.3: Flow of generating a DODAG

First of all, five node topology is created. In this topology, one node is a sink node and the remaining are sender nodes. Then we proceed by clicking on the start button .The actual communication happens between the sink and sender nodes. The intermediate node receives the packet from their child node and at that time calculation of the average queue size of that packet is done by using the below formula as shown.

$$avg \leftarrow (1 - w_q)avg + w_q q$$

If the queue buffer size is 8 bytes, then the minimum threshold is calculated by dividing the queue size by 3 and maximum threshold is calculated by first queue size multiply by 3 then divide by 4 . Then we get the minimum and maximum threshold values for queue size = 8 as 2 and 6 respectively. It is then compared to the average queue size . If the packet is less than these values then it forwards the packet to the parent node. If the average queue size is higher than maximum threshold then

it discards the packet with some probability. In this scenario probability is 0.20000 which is calculated by the following formula:

$$p_b \leftarrow max_p(avg - min_{th})/(max_{th} - min_{th})$$

$$p_a \leftarrow p_b/(1 - count.p_b)$$

The dropped packet is handled by the M-RPL. In this the parent node sends the notification about the congestion via in ACK packet. At that moment of time the node receiving the DIS packet will send the same DIS message to old root node to join tree by comparing the DODAGID. When the root node N receives the DIO message, it checks the DODAGID with the DODAGID that is contained in the DIO message. If the DODAGID is larger than all the DODAGIDs of the node N, which shows the message of a new DODAG formation, the node N restores its DODAGID with received DODAGID and specifies its rank that is equal to received rank incremented by rank 1. If received DODAGID is similar to the node Ns DODAGID, the sender can be seen as another alternate parent node or a sibling node.

Chapter 4

Implementation

4.1 Implementation Tools

4.1.1 Contiki OS

Contiki is open source operation system. It is connected constraint nodes to the Internet. It provides powerful low power Internet communication. It supports IPV6 and IPv4 standard along with 6LoWPAN, COAP, RPL. It can be used in both commercial and non-business and full source code is simply accessible. Contiki application is written is C language and used cooja simulator for simulation purpose so that network can be simulated before burned into hardware.

Features of Contiki

- Memory Allocation

  Contiki is only used for constraint device which is having the limited memory like kilobytes. Contiki is highly memory efficient and provides the mechanism for memory allocation.

- Full IP Networking

  Contiki provides full IP networking support with standard IP protocol UDP, TCP, HTTP and new low power standard 6LowPAN, RPL, COAP.

- 6LowPAN,RPL,COAP

  Contiki support 6LowPAN protocol, RPL IPV6 multi-hop routing protocol and, RPL is network layer protocol.

- Power Awareness

  Contiki basically used for constraint node which low power system. Contiki provides a mechanism for estimating power consumption.

Cooja Simulator

Cooja is network simulator for Contiki which allowed big and tiny network for simulation. Cooja control and feasible study of contiki system via few functions. Cooja is the combination of Java code for the front-end interface and platform specific emulators to carry out the simulations.

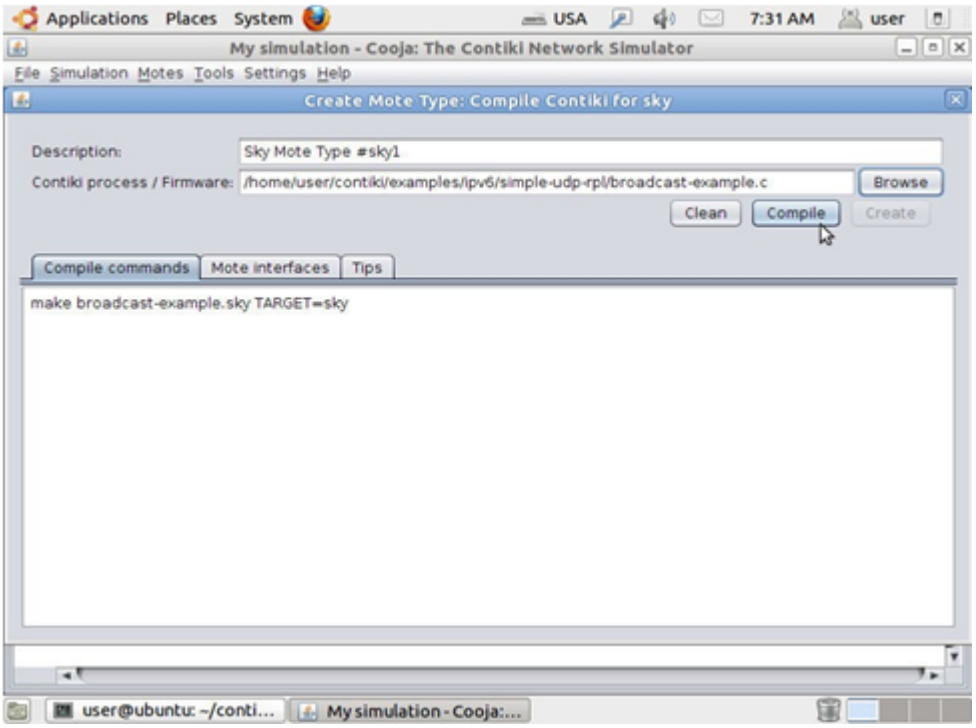

Figure 4.1: Cooja Simulator

Working of Cooja :

First and the foremost step is to download instant Contiki 2.7 version from the In-

ternet and run Instant Contiki2.7.vmx in the virtual machine.

Second step is logging.

Cooja is located in contiki-2.7/tools/cooja. So we have to go to cooja directory.

- Next step is open terminal and goes to specific cooja directory cd Contiki/tools/cooja.
- When double click on an icon of cooja simulator, it will compile automatically.

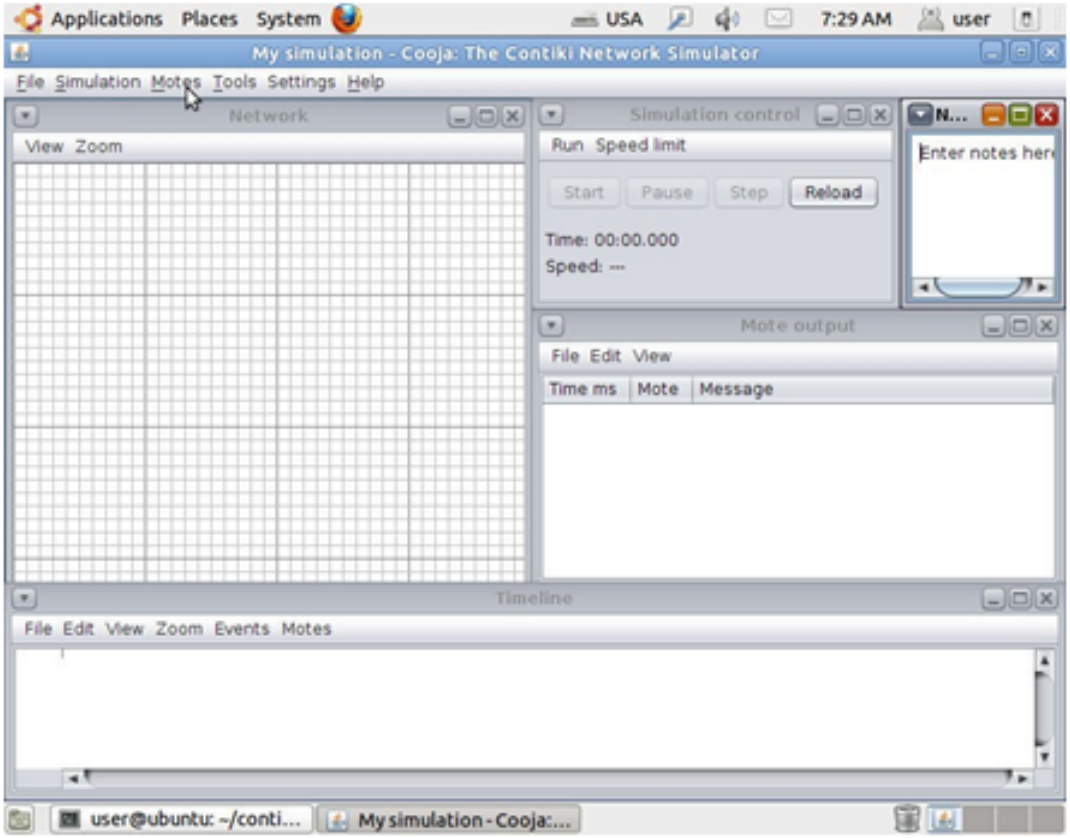

Figure 4.2: Five Windows in Cooja Simulator

- To create new simulator we have run Contiki in cooja simulator.
- Fourth step is to select the file from a menu and click on to New Simulation.

- It opens new simulation dialog box. Type simulation name and then click on create button.
- There is five dialog box are open.
- The network window display the entire node in the entire network.
- Window displays all communication events. The mote output window notifies about the motes id and message related to the mote. In the notes window is a place where we can enter the message for our simulation. The imulation control window is where we begin, stop, and reload our simulation.

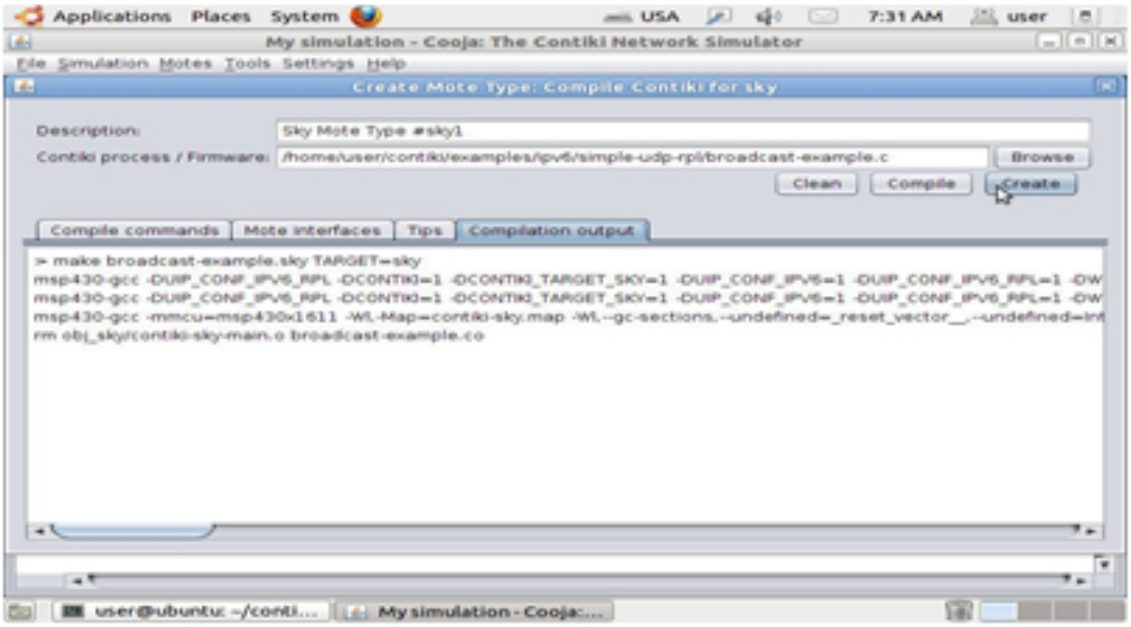

Figure 4.3: Simulatoin of RPL protocol

- To run our application, we must add motes via the motes menu bar.
- Click on Mote, click add mote, click on create new mote type and select any mote from the mote list like we select sky mote.
- Cooja opens the create mote type dialogs; we can change our mote type. Next step is to click browse button to select application from the different type of application

- Here we go to/home/user/contiki/examples/ipv6/simple-udp-rp and open it.
- After selecting application we will compile the code using compile button.
- After completion of compilation process we select create button to create a simulation.
- Cooja will now inquire as to whether we need to include mote from the recently made mote to start the simulation. We change the quantity of bits. Here we include the Number of bits field to 8.

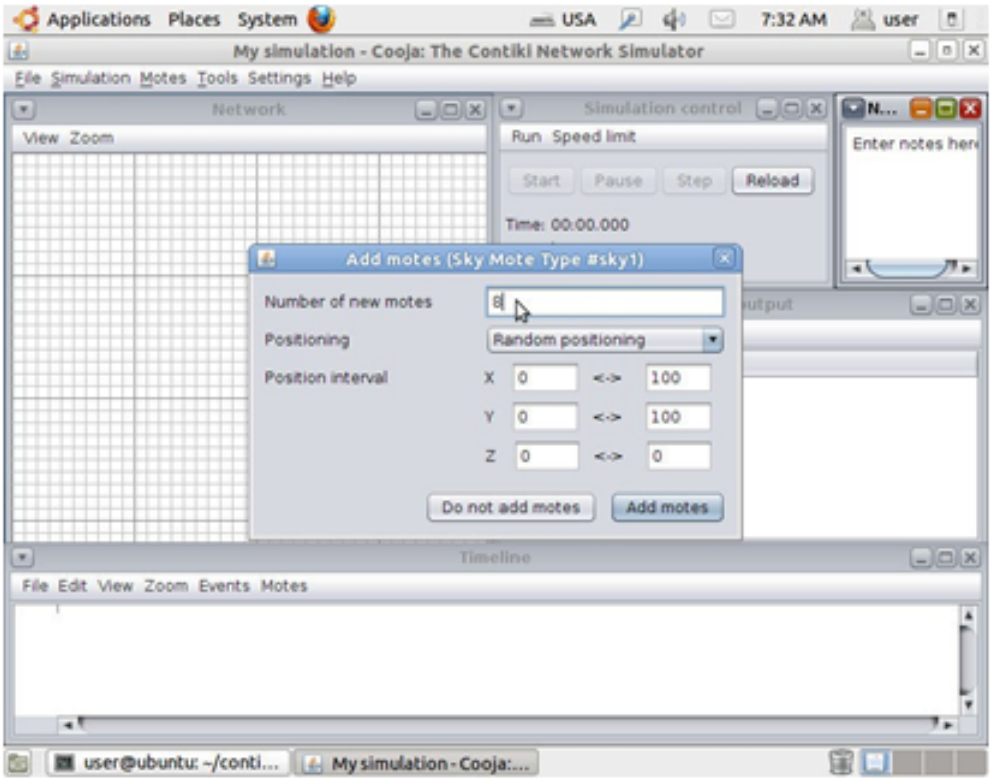

Figure 4.4: Simulation of RPL protocol for 8 nodes

- We See generation from the simulated motes display in the Mote output window. The network window presents communication ongoing in the network. The Timeline window display communication and radio events over time - the small gray lines are Contiki MAC systematically awakening the radio up. We can click the Halt button to pause the simulation [16].

4.2 Results In Contiki OS

Collect View Tool: Collect View tool is one tool in cooja simulator which has the different window that shows the graphical result. First, we have to click send the command to a node to collect the and then to click start collect. Which shows the entire node on the left side. So that we can check graphical result of any node. This tool shows result in different types of graph.

Mote Output Window: The mote output window is used to declare the mote id and time and print-out message. And It is used to display which mote id communicate with other-other mote id and that also display time for actual communication are happen and it shows the print the message on the node it is used to understand node receive the message.

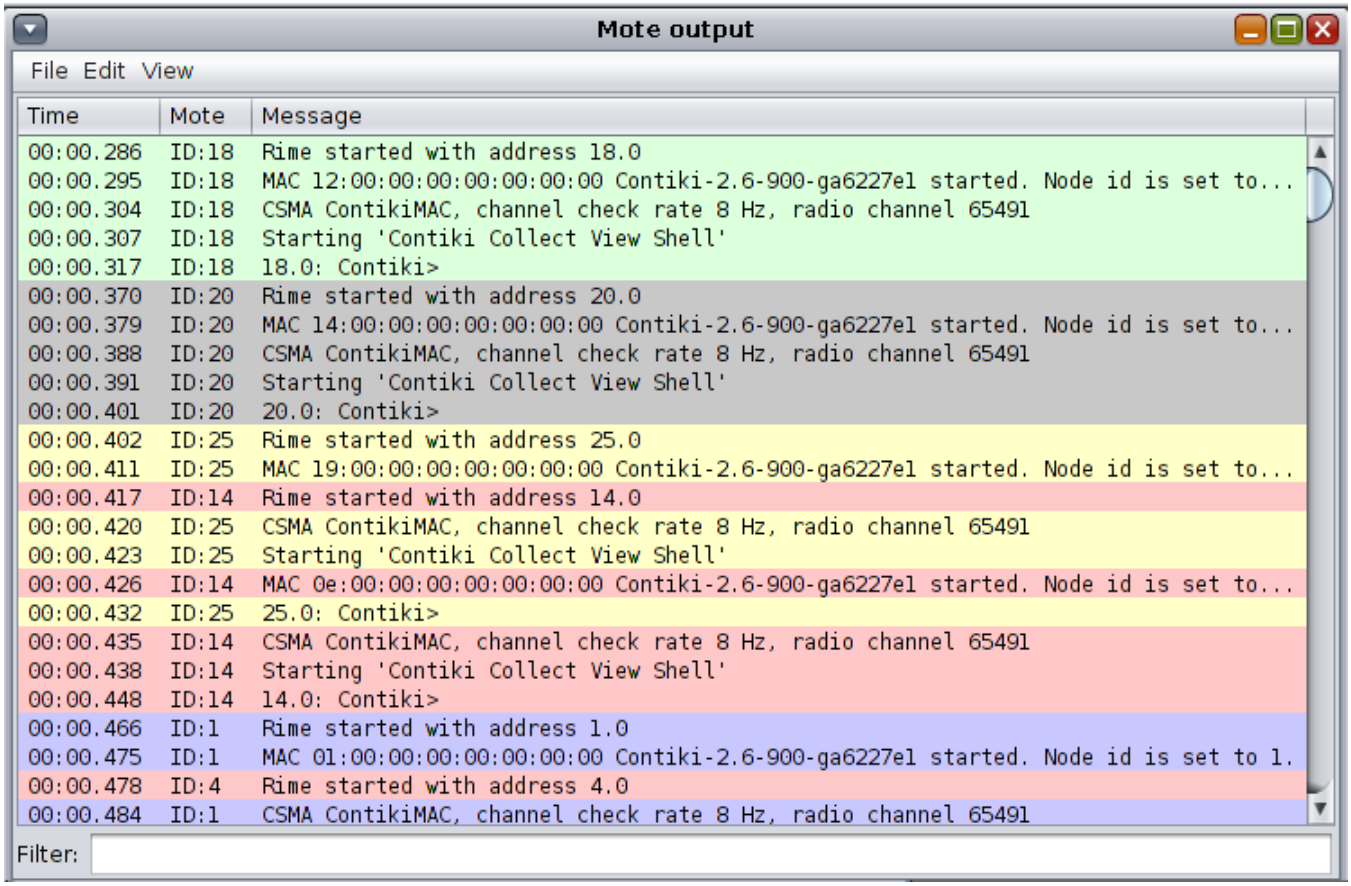

Figure 4.5: Mote Output Window

Node Information Collect: The Contiki cooja provides the inbuilt data collection by clicking on the start data-collect button. That provide different type of information like number of hops require, number of Duplicate packet arrive, lost packet, And how much power is required to transmit data.

Node No.	Received	Hop	Power
1	0	0	0
2	3	1	1.6
3	3	1	2.0
4	4	2	2.7
5	3	3	1.5
6	3	2	1.9
7	4	2	1.9
8	3	2	2.1
9	3	3	1.6
10	3	2	1.6
11	5	4	1.6
12	3	6	1.7
13	4	3	2.2
14	4	4	2.8
15	4	3	1.6
16	3	4	1.2
17	3	3	1.6
18	4	4	1.5
19	5	5	1.7
20	3	4	1.4
21	3	5	1.3
22	4	6	1.4
23	4	5	1.3
24	4	5	1.4
25	3	6	1.4

Table 4.1: Node information

Average Power Consumption: The average power consumption means what is the average power is required to reach a destination. In this figure four types of display one are LPM(live Partition Mobility), Second is the central processing unit, third is the radio listen and last, is radio transmit.

Receive packet per node And Network hops: One other graph shows how many packets are received by the each node. If the node is far away from the source node that time many intermediate nodes have received the packet.

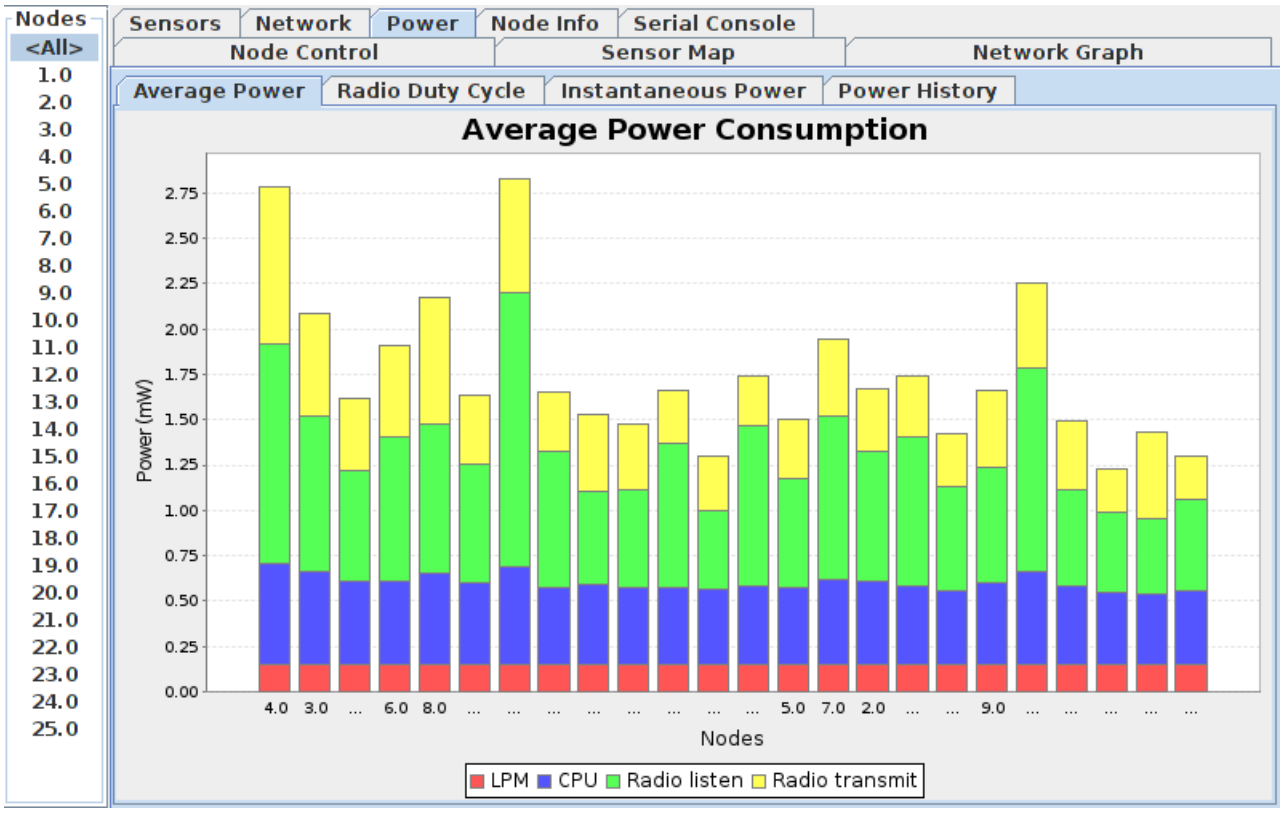

Figure 4.6: Average Power Consumption

The network hops are used to declare the how many hops are required to reach the particular node, if the node is in the transmission range of the sender range then it will directly communicate to the node if the node not in range then required the

multi-hop to reach the particular destination. Other parameter like, Instantaneous power consumption, avg radio duty cycle (radio listen, radio transmit).

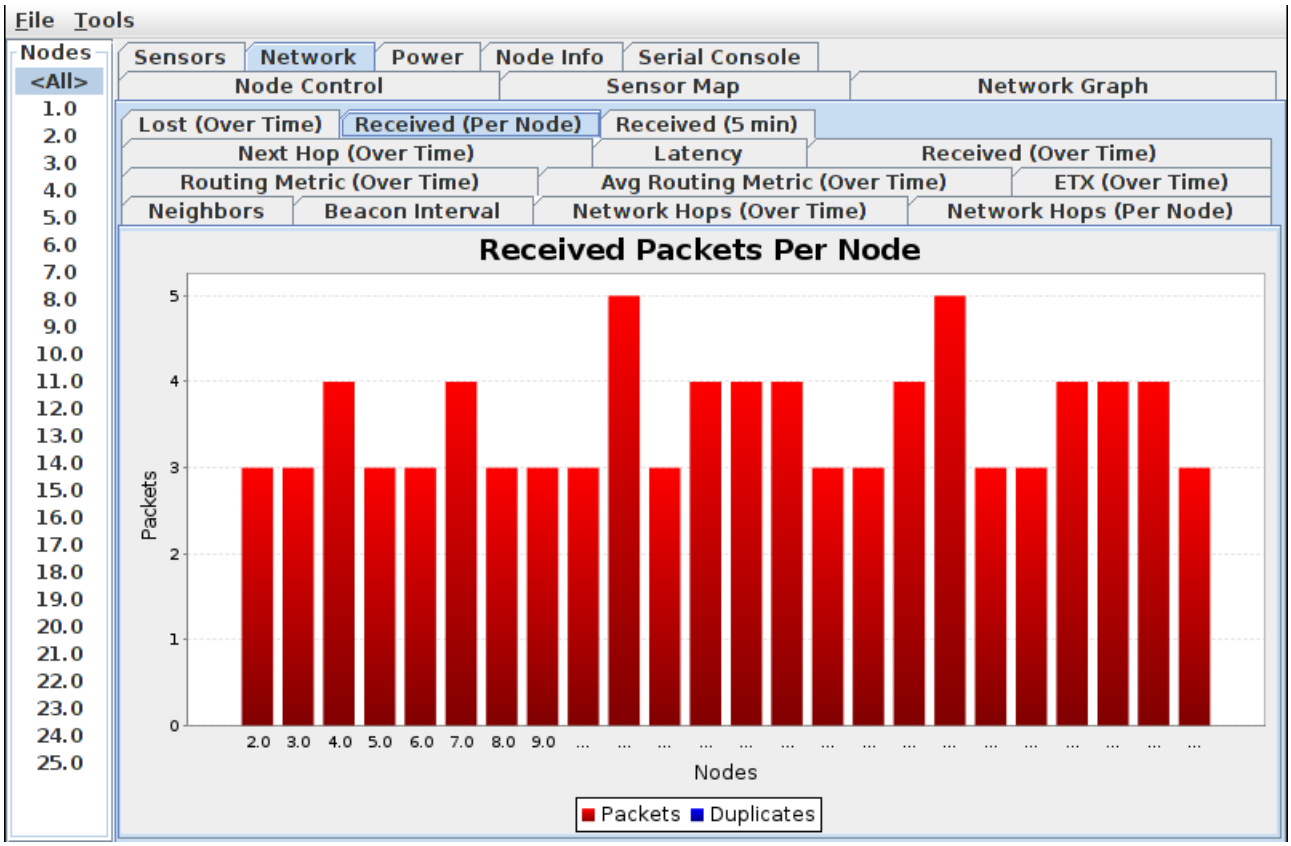

Figure 4.7: Receive packet per node

Chapter 5

Result and Analysis

Congestion occurs when multiple sensor nodes start to send packets concurrently at high data rate or when a node relays many flows across the network. In order to assess the number of lost packets at the wireless channel as compared to at the sensor node experiments using Contiki 2.7 OS and Cooja simulator with different network sizes and various offered loads were performed. These experiments have been executed with and without fragmentation In each network, every node sends packets periodically to a single sink node. The protocol stack and simulation parameters which have been used in the experiments are shown in table. Cooja simulator

Parameter	Value
Simulation Time	**30 Min**
Radio Model	**UDGM**
Node Type	**SKY**
Transmission Range	**50m**
Interference Range	**100m**

Table 5.1: Simulation Parameter

implements a number of wireless channel models such as Unit Disk Graph Medium (UDGM) - Distance Loss which is used in the simulation since interference is considered. In UDGM - Distance Loss, the transmission range is modeled as a disk where all nodes inside the disk can transmit and receive packets. Some protocols,

which need to queue packets, can allocate a queue buffer to store waiting packets such as the MAC protocol that cannot send packets until the wireless channel becomes free. The network sizes are set to be 25 and 50 nodes and the offered loads (packet/second) are 4/1, 2/1, 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, and 1/64. Firstly, we set the MAC buffer size to 8 packets (8 * 127 bytes) which is the default setting of the Contiki OS. Fig. 1 shows the packet loss in 25-node and 50-node networks respectively. Clearly, as offered load and number of nodes increase, the packet loss rises in the network. For example, with an offered load of one packet every second, the packet loss increases from 37% to 76% as the number of nodes in the network increases from 25 to 50.

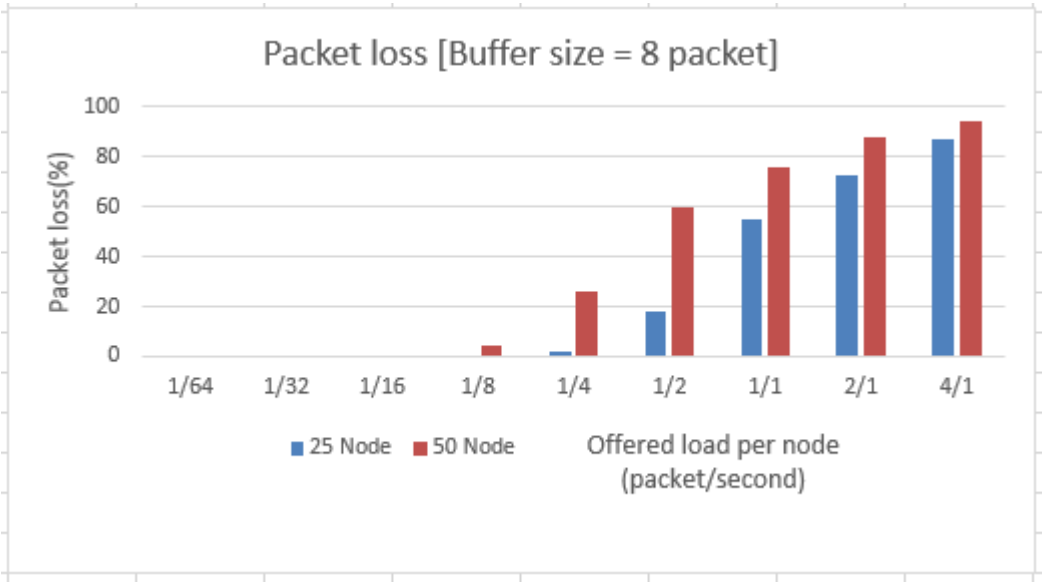

Figure 5.1: Packet loss [buffer size = 8 packets]

The above figure describe the percentage of packet loss. As shown in figure when we send one packet in 64 second then there is no packet loss for 25 node and 50 node topologies. When we send the one packet in 8 second then there is four percentage packet loss of 50 node but zero percentage packet loss for 25 node. When we send one packet in four second then there is two percentage of packet loss for 25 node and 23 percentage packet loss for 50 node and so on.

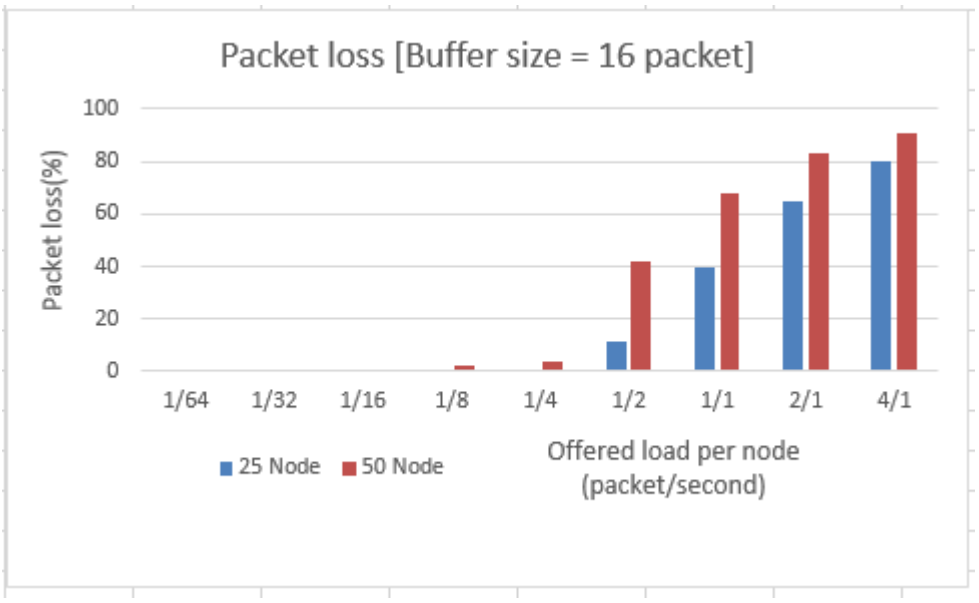

Figure 5.2: Packet loss [buffer size = 16 packets]

Secondly, we increase the buffer size to 16 packets to see the impact of buffer size on the lost packets at the buffer. Figure 5.2 shows packet loss with number of different network sizes and offered loads. By comparing Figure 5.1 with Figure 5.2, it can be seen that by doubling the buffer size, the packet loss decreases with different offered loads. Similarly, the number of dropped packets at the buffer decreases by a small amount.

The graph in Figure 5.1, show the relation between node number and number of hop and number of packets received. In this graph we show that if we send any packet from any node to any node then we estimate the required number of hop and how many packet it can received. So, from this we can reduce the congestion in the network and increase the network lifetime. As see in figure it shows that node 2 received three packets and required one hop and node 3 also received three packets and required one hop and so on.

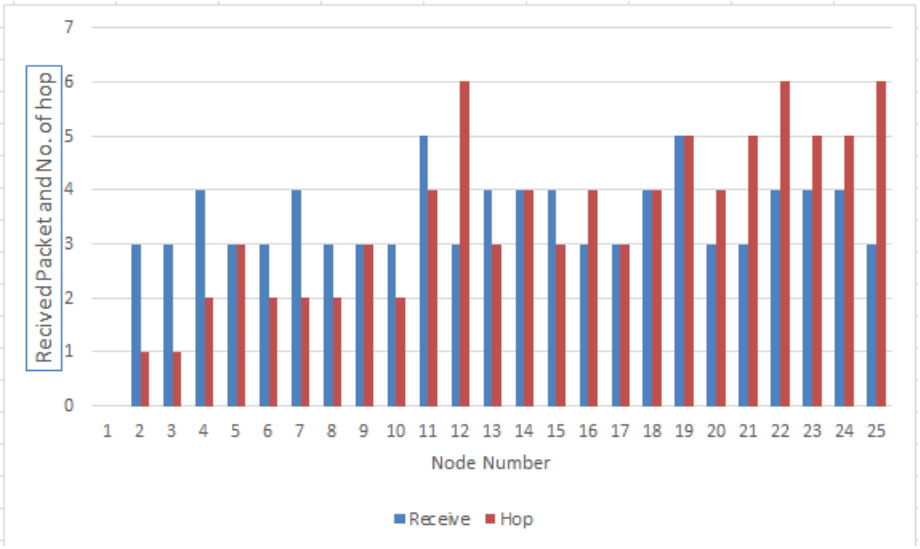

Figure 5.3: Received packet/no of hop

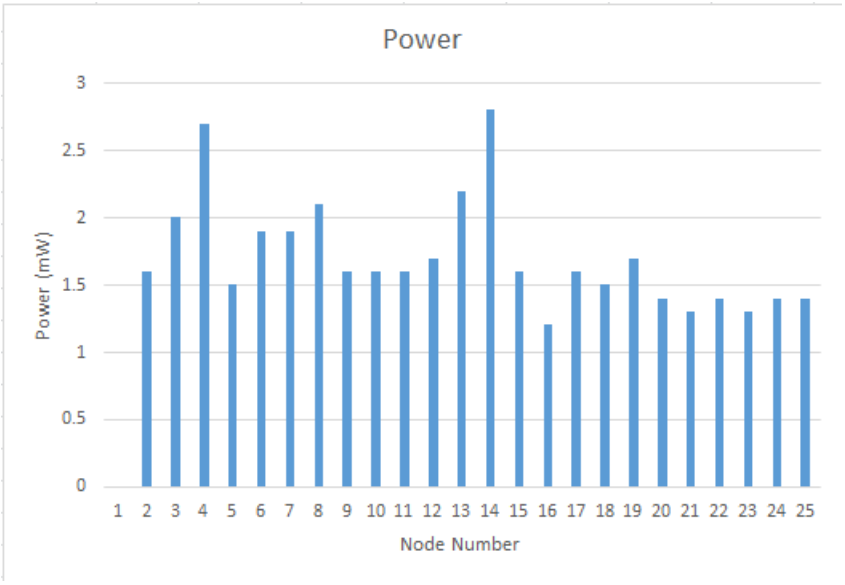

Figure 5.4: p/w consumption by each node

The above graph show the power consumed by sending and receiving packet from one node to another node. Suppose we send a packet from node 2 to node 22 and it takes two hop first it goes node 9 and then goes to node 23 and finally goes to the destination node.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

We can conclude that active queue management can be used in conjunction with explicit congestion warning to adequately degrade packet loss in congested networks. In suitable, Random Early Detection mechanism can provide significant benefits in terms of decreasing packet loss and increasing network utilization. Multipath routing methods are estimated an effective strategy to enhance network potential and resource utilization under difficult traffic situations.

Regardless of whether this approach demonstrates a similar scalability change in genuine arrangement with various radio conditions and topologies and how to reduce the duplicate packet during communication when protocol work under higher activity burdens is as yet open future evaluation.

Bibliography

[1] CASAGRAS, "RFID and the Inclusive Model for the Internet of Things." http://www.rfidglobal.eu/userfiles/documents/FinalReport.pdf/.

[2] M. Rouse, "Internet of Things (IoT) ." http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT/.

[3] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[4] Y.-K. Chen, "Challenges and opportunities of internet of things," in *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, pp. 383–388, IEEE, 2012.

[5] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future internet of things: a wireless-and mobility-related view," *Wireless Communications, IEEE*, vol. 17, no. 6, pp. 44–51, 2010.

[6] J. Vasseur, N. Agarwal, J. Hui, Z. Shelby, P. Bertrand, and C. Chauvenet, "Rpl: The ip routing protocol designed for low power and lossy networks," *Internet Protocol for Smart Objects (IPSO) Alliance*, vol. 36, 2011.

[7] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," 2012.

[8] A. Conta and M. Gupta, "Internet control message protocol (icmpv6) for the internet protocol version 6 (ipv6) specification," 2006.

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

[10] N. Accettura, L. Grieco, G. Boggia, and P. Camarda, "Performance analysis of the rpl routing protocol," in *Mechatronics (ICM), 2011 IEEE International Conference on*, pp. 767–772, IEEE, 2011.

[11] X. Liu, J. Guo, G. Bhatti, P. Orlik, and K. Parsons, "Load balanced routing for low power and lossy networks," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2238–2243, IEEE, 2013.

[12] M. A. Lodhi, A. Rehman, M. M. Khan, and F. B. Hussain, "Multiple path rpl for low power lossy networks," in *Wireless and Mobile (APWiMob), 2015 IEEE Asia Pacific Conference on*, pp. 279–284, IEEE, 2015.

[13] H.-S. Kim, J. Paek, and S. Bahk, "Qu-rpl: Queue utilization based rpl for load balancing in large scale industrial applications," in *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*, pp. 265–273, IEEE, 2015.

[14] C. Kiraly, T. Istomin, O. Iova, and G. P. Picco, "D-rpl: Overcoming memory limitations in rpl point-to-multipoint routing," in *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*, pp. 157–160, IEEE, 2015.

[15] E. Neha, A. Bhandari, *et al.*, "Red: A high link utilization and fair algorithm," *International Journal of Computer Applications Technology and Research*, vol. 3, no. 7, pp. 415–419, 2014.

[16] A. Sehgal, "Using the contiki cooja simulator," *Computer Science, Jacobs University Bremen Campus Ring*, vol. 1, p. 28759, 2013.