Security Analysis of Network Layer Protocol in IoT

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Sumeet Singh (131215)

Under the supervision of

Dr. Shailendra Shukla

To

Department of Computer Science & Engineering and

Information Technology

Jaypee University of Information Technology Waknaghat,

Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled " Security Analysis of Network Layer in IoT " in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2016 to May 2017 under the supervision of Dr. Shailendra Shukla (Assistant Professor (Senior Grade) , Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Sumeet Singh, 131215

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Shailendra Shukla
Assistant Professor (Senior Grade)
Department of Computer Science & Engineering and Information Technology
Dated:

ACKNOWLEDGEMENT

This study has been carried out in collaboration with Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat, Solan, Himachal Pradesh. JUIT is a renowned engineering university in India.

I would like to express my special thanks to all peers and faculty members for their assistance during this study. I am very grateful to Dr. Shailendra Shukla for not only accommodating my queries and tackling various issues but also filling me with inspiration for more improvements in the project work.

I am also thankful to my project partner for his contribution in the critical research work and completion of the project to the current stage.

Table of Contents

Chapter-1 INTRODUCTION

Chapter-2 LITERATURE SURVEY

Chapter-3 SYSTEM DEVELOPMENT

List of Abbreviations

DAO	Destination Advertisement Object
DIO	DODAG Information Object
DIS	DODAG Information Solicitation
DODAG	Destination Oriented Directed Acyclic Graph
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
LLN	Low Power and Lossy Network
MAC	Medium Access Control
OF	Objective Function
RPL	Routing Protocol for LLN
TCP	Transmission Control Protocol
WSN	Wireless Sensor Network
M2M	Machine to Machine
HVAC	Heating, Ventilation and air conditioning
ARP	Address Resolution Protocol
NDP	Neighbor Discovery Protocol
MITM	Man In The Middle
NAT	Network Address Translation
MTU	Maximum Transfer Unit
DAG	Directed Acyclic Graph
ETX	Expected Transmission count
AODV	Ad Hoc On-Demand Distance Vector
DSR	Dynamic Source Routing

List of Figures

List of Tables

Abstract

Internet of Things is a relatively new field that focuses on building networks of devices that are connected over the internet. IoT networks consist of large number of devices. The components of network connected devices normally consists of different kinds of sensors for sensing environment, microcontrollers for processing the data, an energy source (e.g. a battery), protocols and transceivers for data communication. These physical world objects, sensors or actuators, interacts with one another and transfers their detected data to centralized servers. IoT makes use of IPv6 that has 128 bit address space, which basically supports the concept of IoT. IoT has a wide scope for applications for commercialization, but needs several issues to be tackled. The main concern in the field of IoT is security. RPL protocol used at network layer of IoT for routing purposes is vulnerable to many attacks. Our work focuses on realizing the scope of Man in The Middle attack at the network layer of IoT.

Experimental results shows the implementation of Man in The Middle Attack over RPL and proves that security of data in IoT network can be compromised over RPL protocol.

Chapter-1 INTRODUCTION

1.1 Introduction

Deployment of next generation networks has already started due to introduction of new term IoT. Wireless Sensor Network is a group of sensing devices that can interact over the wireless channel where each device can sense, process, and talk to its peers and when these networks gets connected to internet, we call them **I**nternet **o**f **T**hings (IoT).

IoT devices are given unique IP addresses in order to uniquely identify them over the network, where they sense and communicate with each other to achieve common goal. IoT networks usually have a large number of devices due to which they make use of IPv6 i.e. Internet Protocol Version 6 over network layer due to availability of larger address space. IoT has stretched the traditional concept of 'Internet' over laptops and desktops, through mobile and wireless sensor networks. It has changed our perspective of looking at things around us as mere simple items, rather has developed a scenario where these items function as smart objects and aims to increase the quality of our lives.

The components of network connected devices normally consists of different kinds of sensors for sensing environment, microcontrollers for processing the data, an energy source (e.g. a battery), protocols and transceivers for data communication. These physical world objects, sensors or actuators, interacts with one another and transfers their detected data to centralized servers. In Low-Power and Lossy Networks (LLNs) the routers and the interconnected components operates in constrained environment in terms of processing power, memory, and energy. IoT aims to increase the pervasiveness of the Internet by incorporating every single entity for communication via microchip systems, hence leading to the formation of highly complex and dispersed network of human beings as well as devices [2]. IoT is opening wide dimensions for array of applications that focuses on to increase the quality of human lives listed as follows:

- **Smart cities**: Applications in smart parking, traffic jamming, management of waste
- **Smart Environment:** uses in forest fire, earthquake early detection and regulating level of pollution in environment.
- **Smart Water**: can be used detect leakage of water, pollution levels in the sea and rising level of water in rivers.

- **Security and Emergency**: detecting presence of liquid, increasing radioactivity levels and lethal fumes.
- **Industrial Control**: uses in M2M applications, quality of air, HVAC monitoring
- **Smart Agriculture**: for wine quality enhancing, determining levels of greenhouse gases, right conditions for compost.
- **Smart Animal Farming**: grazing, animal tracking, care of offspring
- **Home Automation**: uses in energy and water use, remote control appliances, intrusion detection systems.
- **E-Health**: fault detection, medical fridges, Patient Surveillance

1.2 Problem Statement

There have been a great lot of expectations from IoT and a good lot of applications can be seen in the present world around us. But there is a need to divert our attention towards the security of IoT. There are the following reasons:

1) IoT has stretched the traditional concept of 'Internet', through mobile and wireless sensor networks

2) Anything we can imagine of can be connected to the 'internet', and

3) The connected 'things' needs to interact with each other and this gives rise to greater security and privacy concerns. Confidentiality, authenticity, and integrity of data in the IOT are the major issues to be taken care off.

Increasing number of devices means increasing opportunities for hackers to intrude into the systems and effect the life of normal users. IoT has opened new dimensions for Hackers. IoT Systems basically aims at ubiquitous presence of internet and automation around us. IoT networks have power to fully integrate into our lives and hence their vulnerabilities can have life threatening effects. Some of the scenarios where security of IoT networks is compromised is shown below:

- IoT baby monitor systems can be deployed to keep an eye on the activities of babies to ensure their safety. However if the system gets hacked it can threaten the safety of babies. Hackers can perform various activities like tracking feeds, altering sensor and camera settings and permitting other malicious users to remotely observe and manipulate the monitoring system.

- If security of internet connected cars gets compromised hackers can carry out any number of mischievous events, including taking control of the heating and air conditioning system, unlocking the doors or even shutting down the car in motion.

- Wearables like smartwatches also can become a cause of risk to privacy, where intruders, with the help of motion sensors in smartwatches can steal information you're typing, or they can collect and manipulate health data from smartwatch apps or health tracker devices you might be using.

These just depict the few instances where IoT systems could go wrong and depicts the gravity of situation that security is very important. Quite a lot of methods are already being adopted to prevent security breaches at the device level, and efforts are being made for confrontation of major disasters. Here in our project we will be targeting the security breaches in the IoT network and more precisely at network layer like man in the middle attack in IPv6 using RPL

1.3 Objectives

Our main intentions of this project includes:

- Carry out theoretical study of Man in The Middle attack, RPL protocol, Objective functions, Routing metrics, and IPv6 network and discover vulnerabilities.
- Implement Man in The Middle Attack over RPL routing protocol.
- Analyze the behavior of RPL under Man in The Middle Attack in COOJA simulator.

1.4 Organization

The project report is organized as follows: The next chapter talks about the related work already done and some information required to understand the terms and concepts of work. Chapter 3 presents the development section of the project where experimental model has been shown. Results and analysis performed is shown in chapter 4. Finally chapter 5 concludes the project work and provides some insight about the future work to be done in this field.

Chapter-2 LITERATURE SURVEY

2.1 An Overview of Ipv4

IPv4 is the fourth version of protocol of Internet Protocol (IP). It is one of the main protocol that devices make use of at present times at network layer. It is a connectionless protocol used for packet switched networks and works on best effort delivery model. IPv4 makes use of 32 bit addresses called IP addresses to uniquely identify devices over the network.

2.2 Address Resolution Protocol

For devices to communicate we need both network layer address as well as link layer address. ARP is basically used to map 32 bit IP address to 48 bit machine specific MAC address [5]. The protocol maintains table called ARP cache, which contains the entries of mapped addresses. Protocol makes use of two types of messages called ARP Request and ARP Reply. ARP request message is broadcasted over the network containing the desired IP address to be mapped. ARP Reply is unicasted to the sender of ARP Request containing the MAC address.

2.3 Man In The Middle Attack in IPv4

Man in the middle attack is one of the most predominant attack used against internet users or large organizations. In MITM the attacker node sits in the middle of the source node and destination node and intercepts messages between them [18]. The source node and destination node are unaware of the attacker node and feels that they are interacting directly while in reality communication flows through the attacker node. The final result is that host has access to the sensitive data and it depends upon the attacker node to either manipulate or inject data to get control over the victims.

ARP in IPv4 due to its unsecure functionality is vulnerable to many attacks including Man in the Middle Attack. The attacker node can either respond to the ARP request by faking its identity or can send ARP reply without any request and can forcefully update ARP cache. The hosts feels that it is communicating with the desired host, but in reality it is communicating with the attacker.

2.4 An Overview of IPv6

IP version 6 (IPv6) is the latest version of the Internet Protocol, designed as the successor to IP version 4 (IPv4) [4]. LLN networks make use of IPv6 due extremely large number nodes being deployed in a single network and due to the additional advantages being offered by IPv6 listed as follows:

- **Increased Addressing Space**

  IPv4 addresses were of 28 bit size whereas IPv6 addresses are of size 128 bits, which means additional number of devices over internet satisfying the basic purpose of IoT.

- **Simplified Header**

  Various fields present in IPv4 header are made optional in header of IPv6, hence allowing routers to process header more fast and that too at less cost and exploiting less resources.

- **Concept of Extension Header**

  The options field in IPv4 are replaced by concept of Extension Headers that enables handling additional information in more systematic way.

- **Flow Labeling Capability**

  A new additional feature has been introduced to label the packets belonging to particular traffic "flows" for which the sender requests special handling. It is useful in offering "real-time" services.

- **End-to-end Connectivity**

  IPv6 has finished the purpose of NAT being used in IPv4 and offers end to end connectivity. Every system is allocated a globally unique IP address and hence can be reached directly.

- **Auto-configuration**

  The most important feature of IPv6 is auto configuration of addresses. Just introduce a new device configured with IPv6 into the network and rest of work will be done by device itself.

- **Faster Forwarding/Routing**

  IPv6 header is quite simplified with additional required information being put at the end of header that enables the routers to process the packets faster and make routing decisions quickly.

- **No Broadcast**

  IPv6 does not use broadcast. However it uses multicast to communicate to multiple neighbors.

- **Portability**

  This feature allows users to move in different topographical area and remain connected with the same IP address.

- **Extensibility**

  Options field in IPv4 header is only 40 bytes whereas in case of IPv6 it can be as much as size of IPv6 packet itself. It allows more extra information to be carried.

- **Fixed Header**

  The surprising feature of IPv6 is that though its addresses are 4 times larger than that of IPv4, but its header is only 2 times larger than that of IPv4. The size of header is fixed i.e. 40 bytes and all the extra information is carried by extension headers.

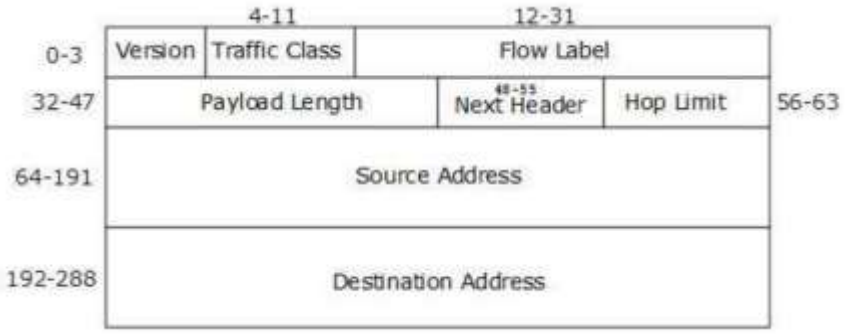

Fig 2.1 Format of IPv6 Header

The various fields of IPv6 header are:

1 Version (4-bits): It depicts the Internet Protocol version and for IPv6 its value is 0110.

2 Traffic Class (8-bits): The most significant 6 bits are used to indicate the routers the type of service the packet is needed. The least significant 2 bits indicates Explicit Congestion Notification (ECN).

3 Flow Label (20-bits): This field is used to label the packets belonging to a particular

flow. It is basically used for real time applications. This allows to arrange the packets in correct sequence.

4 Payload Length (16-bits): This field indicates the size of payload being carried in IPv6 packet. With 16 bits the maximum size that can be indicated is 65536 bytes. However we can carry more data by using the extension headers

5 Next Header (8-bits): This field basically depicts the type of Extension header. It logically links the packets carrying additional information. If the extension header is missing it indicates the protocol being used in upper layer.

6 Hop Limit (8-bits): This 8 bit field is used to prevent the packet to travel for infinite time. It basically limits the packet from travelling specified hops. Its maximum value can be 256. Its value gets decremented each time the packet traverses a hop. This field is same as TTL field in IPv4.

7 Source Address (128-bits): It depicts the 128 bit address of source node.

8 Destination Address (128-bits): It depicts the address of intended recipient of packet

2.5 Neighbor Discovery Protocol (NDP)

IPv6 does not support ARP (Address Resolution Protocol) for resolving IP addresses into MAC layer addresses. ARP in IPv4 is replaced by NDP in IPv6 that does the task of mapping IP addresses to link layer addresses [19]. Discovering of other IPv6 hosts connected on the interfaces and routers is responsibility of NDP.

Some of the major tasks of NDP in IPv6 are:

- Prefix Discovery
- Neighbor Discovery
- Router Discovery
- Duplicate Address Detection (DAD)
- Address Resolution

In order to perform their functionalities NDP makes use of 5 types of ICMPv6 messages. ICMPv6 messages are used for error reporting and network diagnostic functions [5]. ICMPv6 is a vital element of IPv6, and every IPv6 node must implement it.

2.5.1 General Message Format of ICMPv6

The ICMPv6 messages have the following general format:

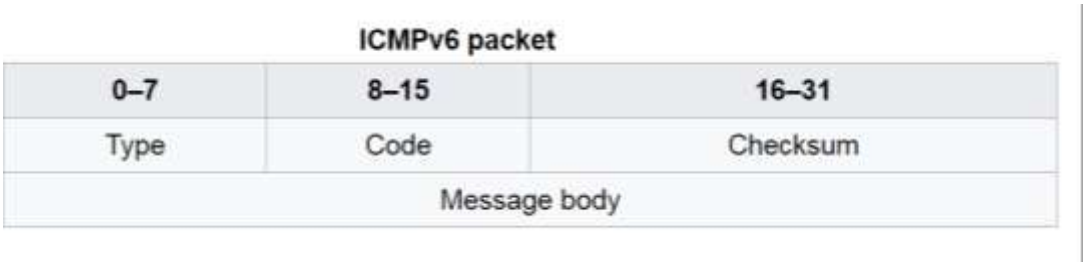

Fig 2.2 ICMPv6 packet format

The **Type field** indicates the type of the message. The format of the remaining data depends upon this value.
The **Code field** provides additional information and depends on the message type.
The **Checksum field** is used to detect data corruption in the ICMPv6 message and parts of the IPv6 header.

2.5.2 ICMPv6 Messages

Neighbor Discovery makes use of 5 type of ICMPv6 messages:

Neighbor Solicitation (Type 135): An IPv6 configured node makes use of neighbor solicitation message to resolve the layer 2 address of its neighbor located on the link or to verify the earlier known link 2 addresses.
Neighbor Advertisement (Type 136): This message is generated in response to the neighbor advertisement message and it contains the link layer address of sender. However unsolicited neighbor advertisement messages are generated to inform changes in link layer address of node.

Router Solicitation (Type 133): These messages are multicasted by IPv6 configured hosts to solicit the routers to send the configuration information. However, routers usually send required information periodically, but it forces routers to respond immediately.

Router Advertisement (Type 134): IPv6 routers send router advertisement messages in response to router solicitation messages and these messages contains the information required by hosts like link prefixes, link MTU and hop limits.

Redirect (Type 137): Redirect messages are sent by routers to inform nodes of better next hop Routers.

2.6 Man In The Middle Attack In IPv6

Functionality of ARP in IPv4 is performed by Neighbor Discovery Protocol (NDP) in IPv6 [20]. NDP in IPv6 is a victim of similar kind of vulnerabilities as ARP in IPv4. Man in the middle attack in IPv4 could be implemented in various ways such as ARP cache poisoning and DHCP spoofing. Similarly in IPv6 Man in the Middle attack could be performed by following ways:

- Man in the middle with spoofed ICMPv6 neighbor advertisement.
- Man in the middle with spoofed ICMPv6 router advertisement.

.

2.6.1 MITM with Spoofed ICMPv6 Neighbor Advertisement

Neighbor discovery in IPv6 is performed by making use of two types of ICMPv6 messages, ICMPv6 neighbor solicitation message and ICMPv6 neighbor advertisement message. These two messages are used to resolve the IP address into MAC address. Fig 2.3 below shows the normal process of IPv6 discovery in the network

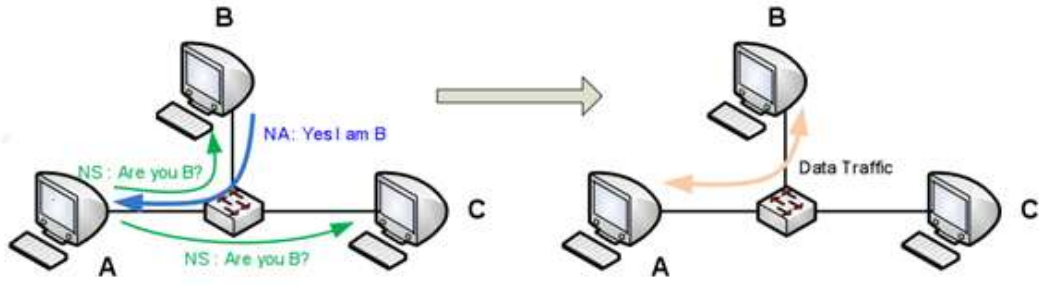

Fig 2.3 IPv6 Discovery

In Fig 2.3 above, NS depicts the ICMPv6 neighbor solicitation message and NA depicts the ICMPv6 neighbor advertisement message. If Node A desires to communicate with Node B it will carry out carry out series of steps. First Node A will multicast ICMPv6 NS message for all nodes. Every node on the network will receive this message. Node B will process this message and in return will send NA message with solicited flag to Node A. Node A will successfully learn the MAC address of Node B and hence the communication will start between the nodes.

However the process of resolving IP address in IPv6 is similar to that of Address Resolution Protocol in IPv4 and hence this process suffers from same vulnerabilities.

Fig 2.4 shows how MITM attack can be implemented in this case.

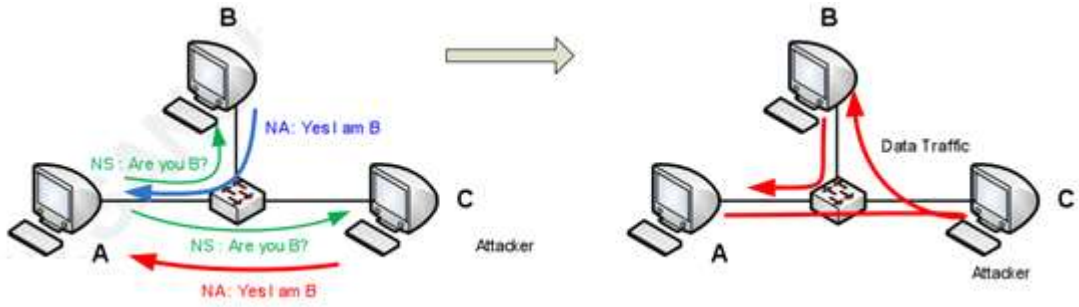

Fig 2.4 Man in the Middle

When Node a wants to find the MAC address of Node B, it will multicast NS message for all nodes. All nodes present in the network including Attacker node will receive this message. Node B will respond to NS message with NA message with solicited flag enabled. However in this case Attacker node will also respond to NS message with both solicited flag and override flag enabled. Override flag will override the existing entry for Node B in neighbor cache table. Node A will start communicating with attacker node, assuming that it is interacting with Node B. Now Attacker node has options to conduct further attacks.

2.6.2 MITM with Spoofed ICMPv6 Router Advertisement

When a Node joins a network it sends ICMPv6 router solicitation message to communicate with routers. The routers in response generate Router Advertisement message containing network prefixes, options, lifetime and auto config flag.

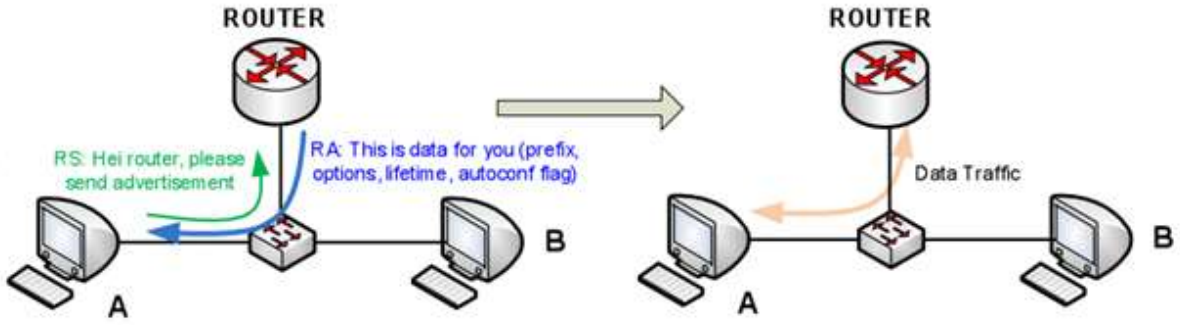

Fig 2.5 Router Advertisement

Fig 2.5 shows normal functionality in which a Node A generates RS message and multicast to all routers. Each router present in the network will get this message. Router process this message and multicast Router Advertisement message to all nodes in the network. Node A receives the required information, configures its routing table and start its functionality.

However the issue is that any node can claim to be router. Fig 2.6 shows how MITM can be realized in this scenario.

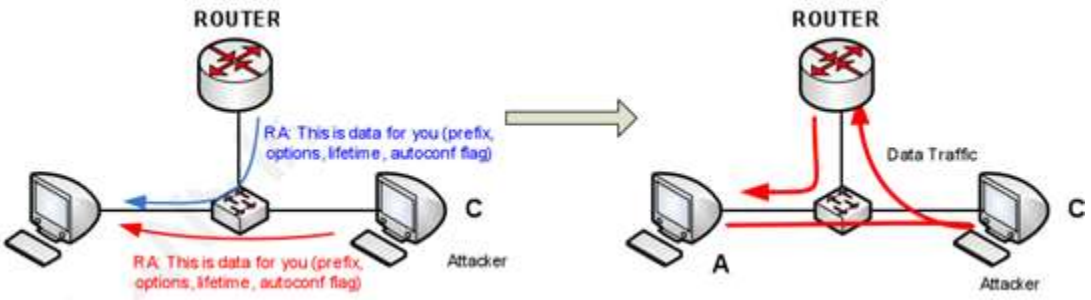

Fig 2.6 Man in the Middle

Router generally sends Router Advertisement message over the network so that the nodes can configure their routing tables. Attacker can send ICMP RA message over the network and announces itself the router with highest priority. Nodes in the network can select malicious router as a default gateway and all the traffic meant for outside network will pass through the attacker.

2.7 Overview of RPL

The main purpose of routing protocols is to forward the packets form one source to destination by selecting specific routes. We basically witness 2 types of routing protocols in Wireless sensor networks: Reactive and Proactive [8]. The reactive protocols determine routes when required. Examples of this type of protocols are AODV [9], DSR, and TORA [10]. However average end to end delay will be more as time is consumed in determining path every time. On the other hand proactive routing protocols determines the routes in advance so that the path is already present before needed due to which these protocols exchanges control messages from time to time to determine and select the routes in the network in advance.

Routing protocol for low power and lossy networks (RPL) is IPv6 routing protocol indicated by IETF which put forwards the routing mechanism in Internet of Things (IoT). RPL is a distance vector routing protocol as linked state routing protocols makes use of substantial amount of memory and resources which are not available in the resource constrained LLNs. Being a proactive routing protocol, RPL starts the discovery of routes as soon as the RPL network is initialized.

RPL fundamentally works with the development of a tree-like topology called Directed Acyclic Graph (DAG) in which no loops are present. Every node in a RPL tree has a favored parent which functions as a gateway for that node. In the event that a node does not have a passage in its routing table for a packet, the node just advances it to its favored parent until it either gets delivered to the destination or a typical parent which advances it down the tree towards the destination. The nodes closer to root have bigger routing tables bigger as the nodes in a RPL organize to have a path for every one of the nodes down the tree.

The main purpose of RPL is to find the paths from each node to the sink and is done by the use of routing metrics, objective functions and routing constraints.

Internet Protocol (IP) offers end to end connectivity and RPL makes utilizes TCP/IP for communication. It also eradicates the issue of interoperability between the devices from different vendors [11], [12]. It also enables the growth of applications and integrations in terms of data collection and configuration [13]. Being a lightweight protocol we can expect IP to function on even systems with less resources [14].

Topology Formation

Since LLN networks works on wireless channels, so they do not have predefined topologies, as determined by wired links, so RPL needs to find connections and neighbors to frame a topology first.

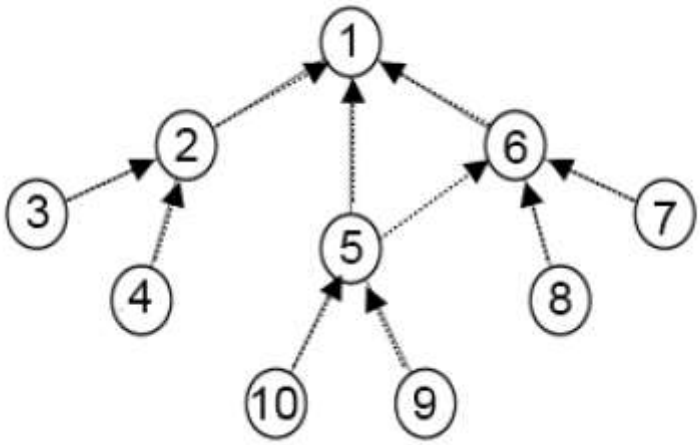

Fig 2.7 Example of topology created in RPL

RPL builds a tree like topography as shown in fig 2.7 with root at the top and leaves at the edges. In comparison to tree topology, extra links are also offered by RPL which is a must requirement in LLN [9]. Movement of Traffic in RPL is depicted in two ways using "up" and "down" direction's terminology. The down from the root to the leaves and up is from the leaves to the root.

2.7.1 Routing in RPL

RPL is a distance vector protocol and working of the protocol can be observed into 2 main segments.

1. Routing upward
2. Routing downward

RPL organizes its structure in the form of Destination Oriented Directed Acyclic Graph (DODAG). The DODAG holds the routes from the leaves nodes to the root. The root can be known as LLN border router (LBR) and has no outgoing edges. The root may be connected

to a private network or any other non-LLN network, in which case the DODAG is considered to be grounded. If not connected DODAG is considered floating.

Routing Upward

In RPL upward routes are constructed to facilitate the traffic flow from leaf nodes to root node. Each node is assigned a preferred parent and the DODAG informs who the parent of the node is. So to forward the packet to the root each node forwards the packet to its parent and this process continues until the packet reaches the root. RPL makes use of ICMP control message called DODAG Information Object (DIO) to build the upward topology.

RPL makes use of the idea of ranks being allotted to each node. Rank basically determines the relative position of node in graph from the DAG root and ensures that no cycles are present in graph. Rank of node increases as we move away from root and decreases in the opposite direction and node cannot join parent with higher rank. Rank calculation in the DODAG is based on the definition of Objective Function (OF) that guides the DAG construction in RPL. Objective Function takes into account some routing metric eg. ETX (Expected Transmission Count) that considers the link quality between the nodes before selecting the path and nodes select parent with minimum ETX. Lower rank neighboring nodes acts as candidate parent of node and the node selects one of the candidate parent as its preferred parent by choosing parent with lowest rank and best routing metric (ETX).

Routing Downward

Downward routes are required to facilitate point to multipoint (P2MP) communication i.e. from root to leaf nodes. RPL makes use of ICMP control message called DODAG Destination Advertisement Object (DAO) messages to form the routing table for downward traffic. Though downward traffic indicates movement from root node to leaf node DAOs are always sent in upward direction. Once the upward routes have been formed the transmission of DAO messages begin.

2.7.2 RPL Messages

RPL makes use of 3 types of ICMP control messages for generating RPL topology and maintaining routing table.

These messages are: DODAG Information Object (DIO), DODAG Information Solicitation (DIS) and DODAG Destination Advertisement Object (DAO).

1. **DODAG Information Object (DIO)**

   DIO messages are used by protocol to form upward routes. DIO messages move in downward direction and facilitates multipoint to point communication. DIO messages basically contain information about DODAG that helps nodes to discover and join the DODAG and select the parents. As soon as the network starts the nodes starts multicasting DIO messages to their neighbors to form the topology.

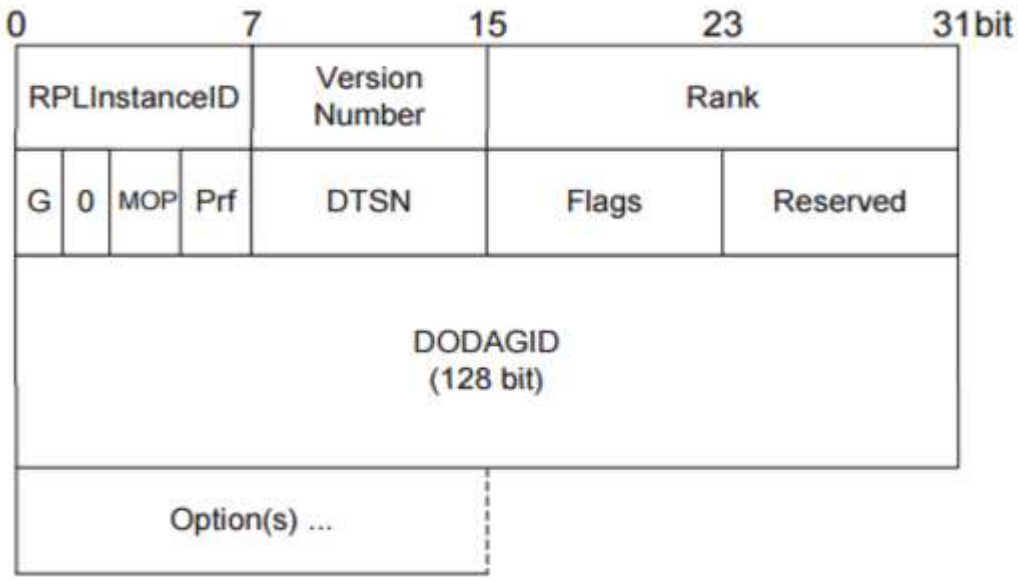

Fig 2.8 DIO Message Structure

2. **DODAG Information Solicitations (DIS)**

   The DIS messages are basically used to explicitly solicit nodes to send DIO messages. These are used in cases when a new node joins a network or when it fails to receive DIO after specific wait.

3. **DODAG Destination Advertisement Object (DAO)**

   The DAO messages are send by RPL nodes to form downward routes. DAO messages travel in upward direction. These messages are unicasted from child node to

its parent node. Once the upward routes have been formed the transmission of DAO messages begin.

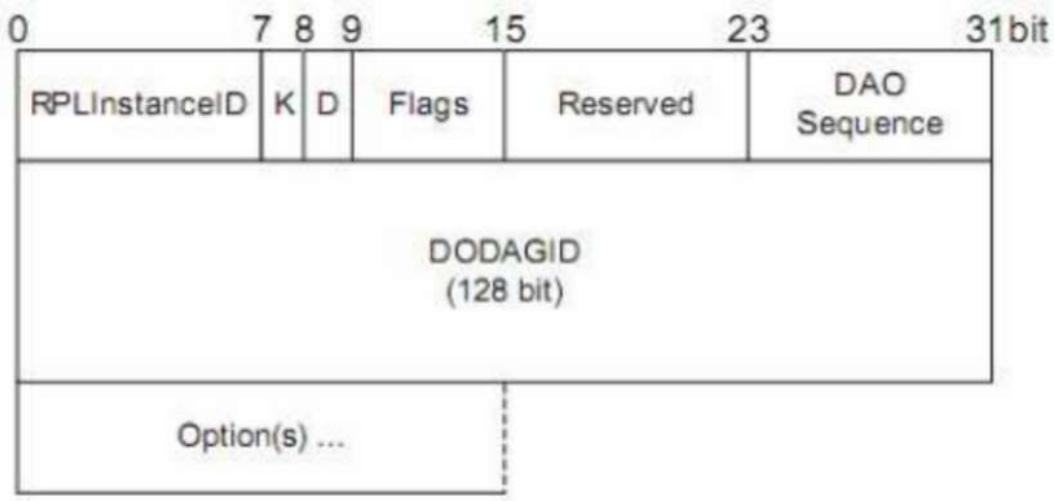

Fig 2.9 DAO Message Structure

2.7.3 Objective Function

The main driving factor of RPL is the objective function. An objective function determine how the nodes will calculate ranks and select their parents in order to optimize routes. The objective function makes use if some routing metrics or constraints to make the decisions. A routing metric is generally a measureable value used to determine the cost of path and allows to select a particular path in case there is an option of multiple paths. An application may require a fast delivery of packets using a shortest path and hence the aim will be to minimize ETX metric. A node may be low on energy and the target will be to increase the duration for which network works by using as many mains associated nodes and ignoring battery operated nodes along the path. So objective function is selected according to different scenarios and different application needs. Objective function is determined by OCP (Objective Code Point) in DIO messages. ContikiRPL implements two OFs i.e. OF0 and ETX. OF0 uses hop count as routing metric where as ETX uses ETX metric as a routing metric for selecting the best path [9].

2.8 Decrease Rank Attack

RPL protocol is new protocol and hence suffer from various vulnerabilities. The existing literatures talks about various attacks being performed on RPL and one of them is decrease rank attack [21]. Rank property in RPL seems to be very sensitive and it can be manipulated to effect the normal functionality of protocol. In decrease rank attack, the attacker node deliberately changes its rank to lower value and could attract the traffic towards itself. This attack can be performed to implement other attacks also such as wormhole attack, sinkhole attack etc. However the performance of attacker node gets compromised as attacker has to handle additional traffic after the implementation of attack.

2.9 Our Contribution

Man in The Middle attack appears to be very significant attack and its scope is clearly visible at the network layer of IoT that makes use of RPL protocol for routing purposes. The attack gives options to attacker to do whatever he wants to do with data, either simply read data or manipulate it. It even questions the availability of data. The worst part of attack is that theoretically it does not cause any visible changes in the network and silently performs its tasks. We in our project tends to focus on implementation of this attack on the network layer of IoT by exploiting the rank property of RPL nodes and analyses the effects and results of this attack.

Chapter-3 SYSTEM DEVELOPMENT

3.1 Contiki, Sensornet Operating System

Contiki is an open source operating system for networked devices that works in constraint environments in terms of memory and processing power. Contiki includes several libraries, kernel, program loader and various processes [8]. This operating system basically resides in smart objects.

Contiki is basically used for programming the smart object applications. It makes use of IP communication provided by cisco. The operating system and its applications both are developed in c, and hence the applications are highly portable to variety of architectures

Contiki is an event-driven operating system in which processes are executed as event handlers that aims to their completion. The main features of contiki include coffee file system, multiple hardware support, light weight communication stack, protocol support like 6lowpan, coAp, RPL etc. [8]

Important aspect of contiki operating system is that it is open source OS freely available for download with the name of InstantContiki and requires no tedious setup.

The Contiki operating system have multiple modules located in multiple files. Different routing modules and files are located in a directory “contiki/core/net/rpl. These files are separated on the basis of functionalities they provide for instance rpl-dag.c contains the functionality for formation of Directed Acyclic Graph (DAG), rpl-mrhof.c and rpl-of0.c talks about the objective functions being used in contiki, rpl-icmp6.c controls behavior of control messages etc.

In this study most of the work is related to these files of the Contiki operating system

3.2 Cooja Simulator

Cooja is a simulator provided by contiki operating system developed in java and is basically used for simulating sensor networks. [9].The simulator makes use of sensor nodes programmed in c.

The main feature of cooja is that it provides several tools helpful to observe the actual behavior of network behavior. Its collect application provides various features like sensor map to visually observe the RPL tree formation. Node info tab provides all information necessary like packets sent by each node, loss of packets, value of routing metrices, power consumption and duty cycle of each node etc.

It saves the information about the simulations in xml file with extension 'csc' (Cooja simulation configuration). The radio logger saves the output of the nodes with time stamp and which can be saved in file also for future references.

3.3 Setting up Environment

3.3.1 Creating Simulation

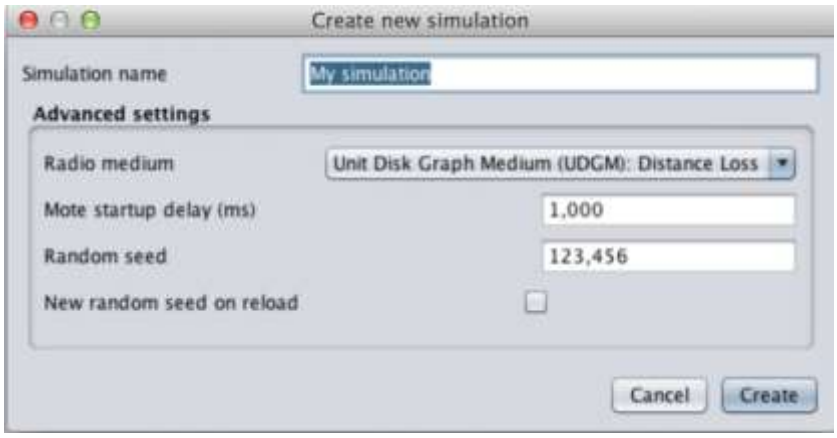

Fig 3.1 Create new simulation interface.

The process starts by making a new simulation. First, click “File > New Simulation”. The dialog box will appear as shown in fig 3.1.Then provide the simulation with a suitable name and select the radio medium. In most cases, Unit Disk Graph Medium (UDGM) is quite suitable. Random start-up allows the motes start at random times to prevent starting them at the same time. Random number generation requires a seed provided by main random seed. Check the box to start motes with random seed. Now click "Create" to create simulation after which the interface presented in fig 3.2 is shown.

3.3.2 The Simulation Interface

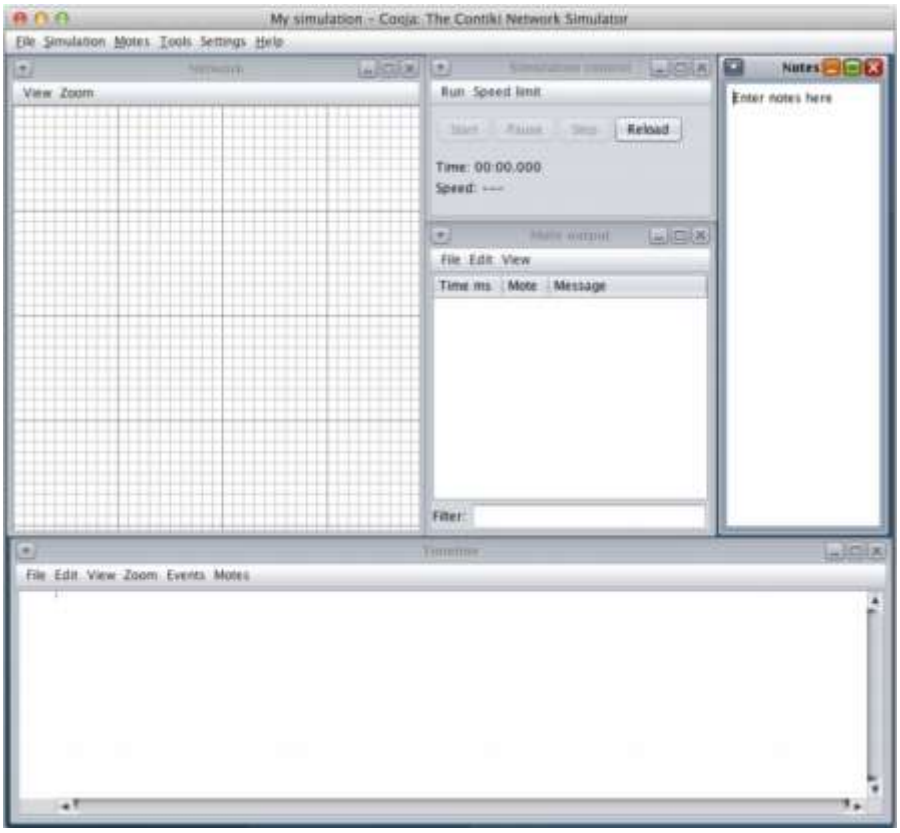

Fig 3.2: The Simulation Interface.

There are five windows in the simulation interface shown in fig 3.2. First is the Network window that depicts the actual outline of the network. It allows to physically move and place the nodes and make topology as required. The Simulation Control window enables to start, reload and stop the simulation. The speed of the simulation could also be controlled by this window. The Mote Output window acts as a console where motes print their outputs. The "Filter" field allows to filter out the results based on the entered text. The Timeline window shows various events like LED activity, radio traffic or anything that occur on each mote over the period of simulation. The Notes window is used for making temporary notes in the simulation.

3.3.3 Selecting Type of Mote

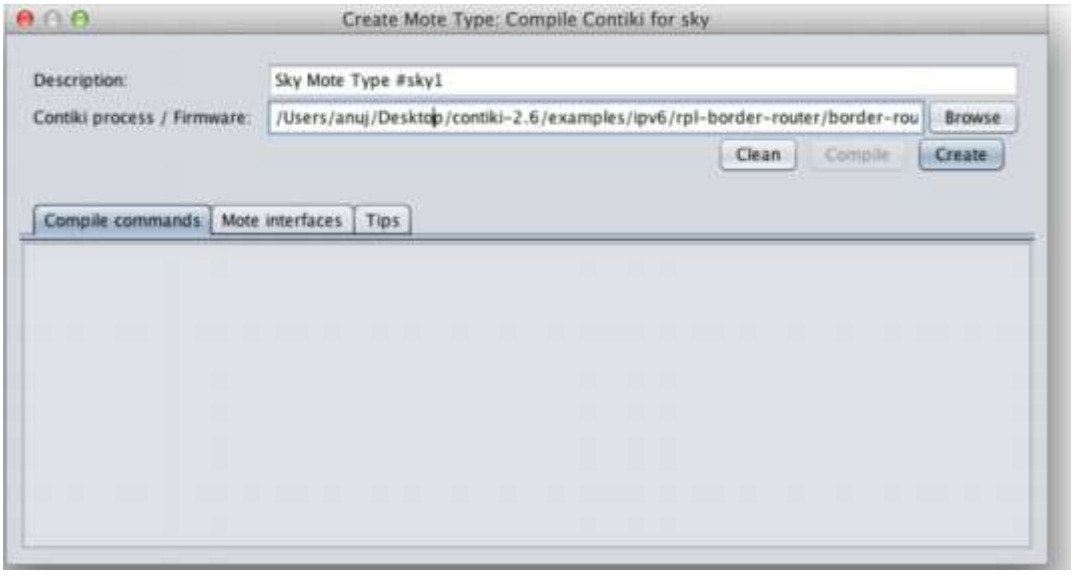

Fig 3.3 The Create Mote Type Interface

The further step is to select the type of mote types. Cooja offers motes of various platforms. These motes differ in terms of memory and range. We start by creating the mote types. Click "Motes > Add Motes > Create New Mote Type > Sky Mote". Specify the suitable description for the mote. Add the description of first mote be server mote as shown in Fig 3.3. In the "Contiki Process / Firmware" field, either specify the binary file (the .sky file) or the source code file (the .c file). Binary files needs no compilation, they are formed after compilation. If source file is specified the compile button becomes active. Press the compile button to compile the file. The output of the compilation is visible in the compilation output tab. If successful, the Create button becomes available. If "Add Motes" window is shown, please click "Do not add motes" for the moment. In this way create the required motes. For our simulation we make use of two types of motes with source files udp-sender.c (client) acting as nodes sending data and udp-sink.c (server) acting as sink for network. These files are located in folder home/contiki/examples /ipv6/rpl-collect.

3.3.4 Adding Motes

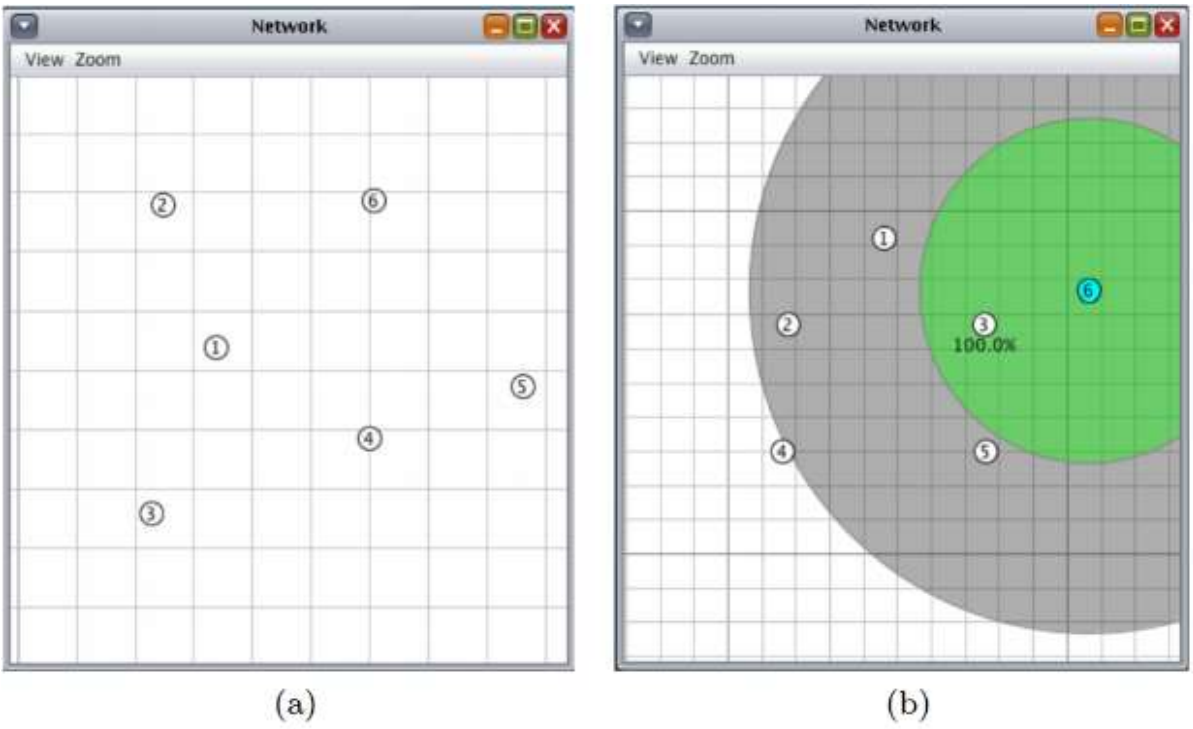

Fig 3.4 The Simulation Interface

Now after the desired motes have been created now the motes can be added in the environment. Click “Motes > Add Motes” in the menu bar. It will show the names of motes already created. Add 1 server mote and 29 client motes. We can either position the motes manually by specifying the coordinates or choose random positioning. After this the total of 30 nodes will appear in the network window. These motes can be dragged to build the required topology as shown in fig 3.4 (a). Turn on the radio environment by clicking on the view option in network window. Click a node to view its radio environment.is clicked it also shows the radio environment in green and grey color. The green circle represents the area of successful transmission and reception range of the nodes and grey area represents the part where interference from other nodes can exist. Fig 3.5(b) represents an example where maximum of two nodes can exist within range.

3.4 Initial Sample Simulation of the Topology

Fig 3.5 depicts a sample network that contains 1 server node and 6 client nodes. Node 1 will be acting as a DODAG root of RPL tree. The server node is using the file udp-sink.c and the client nodes will be using the file udp-sender.c, both located at location home/contiki/examples /ipv6/rpl-collect. We make use of Cooja Unit Disk Graph Medium in order to introduce lossyness with respect to relative distances of nodes in the Radio medium.

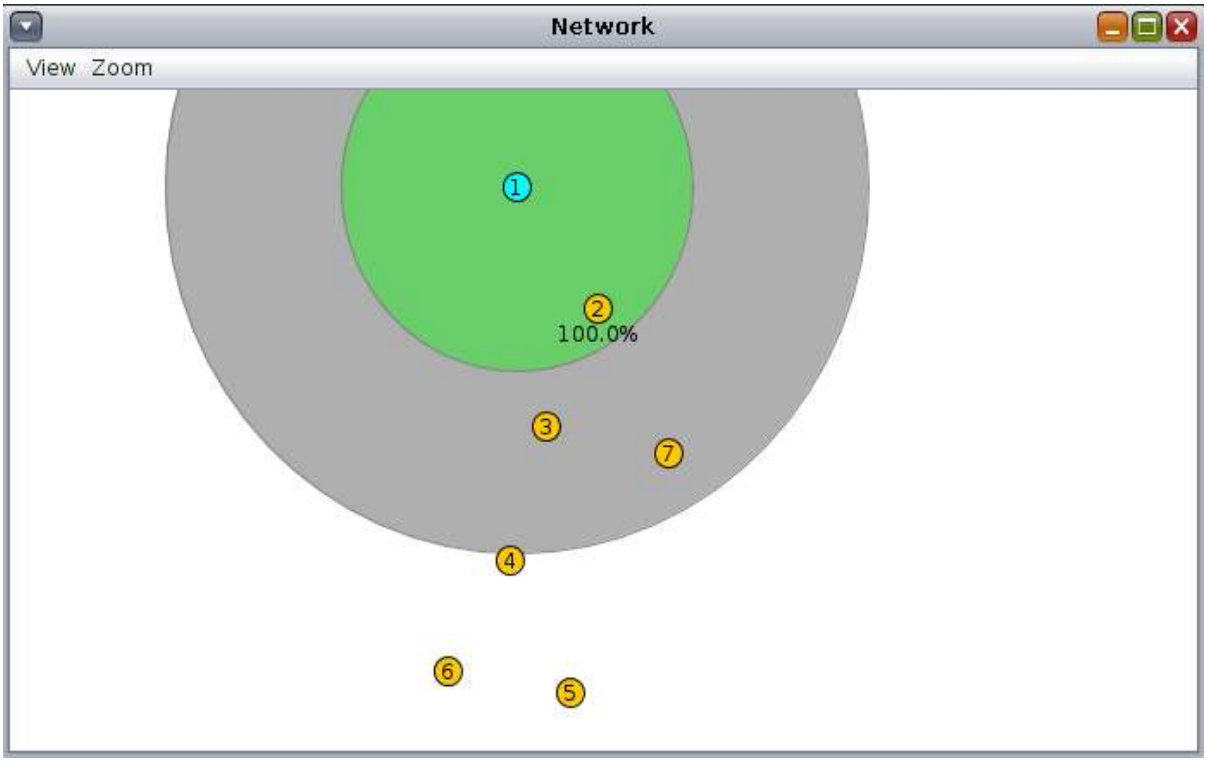

Fig 3.5 Sample simulation with 6 client nodes and 1 sink node

The blue node numbered as 1 is the sink node and yellow nodes are the client nodes that send packets to sink node.

To run a simulation click Start.

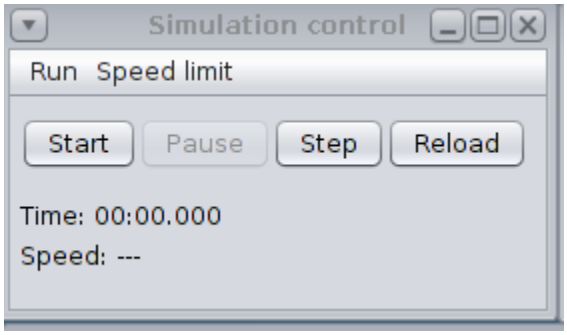

Fig 3.6 Simulation control

The fig 3.7 shows simulation in the running phase:

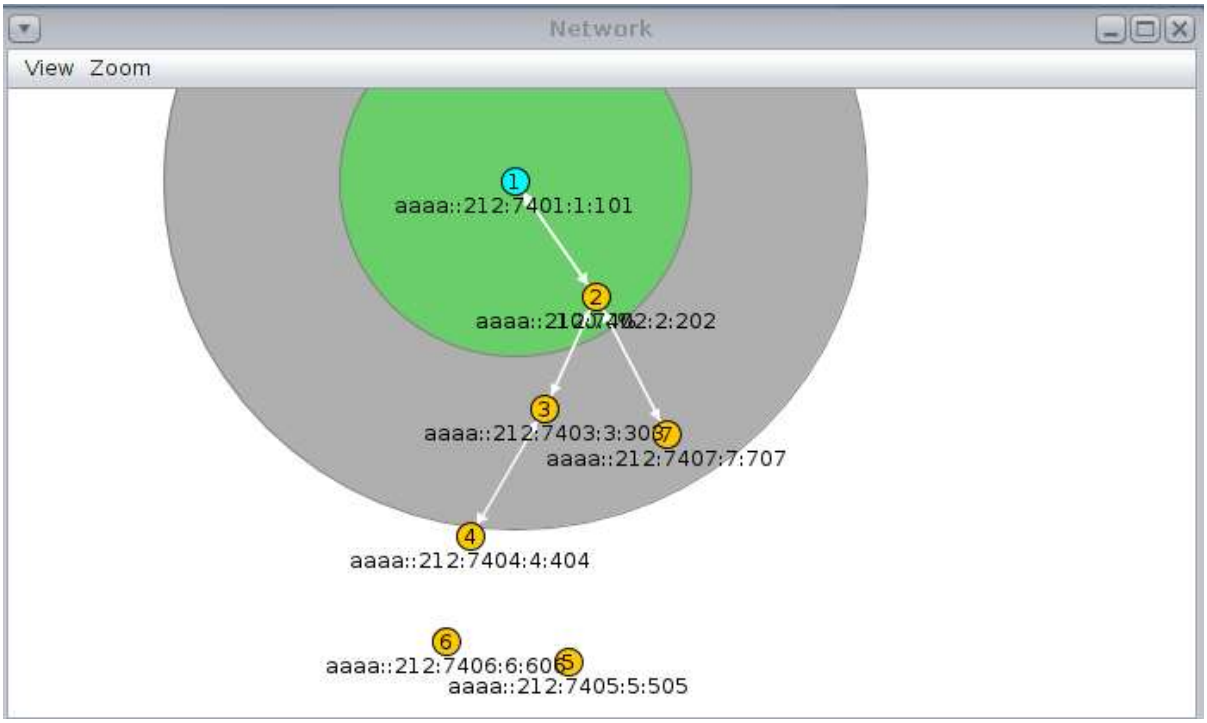

Fig 3.7 Simulation in running phase

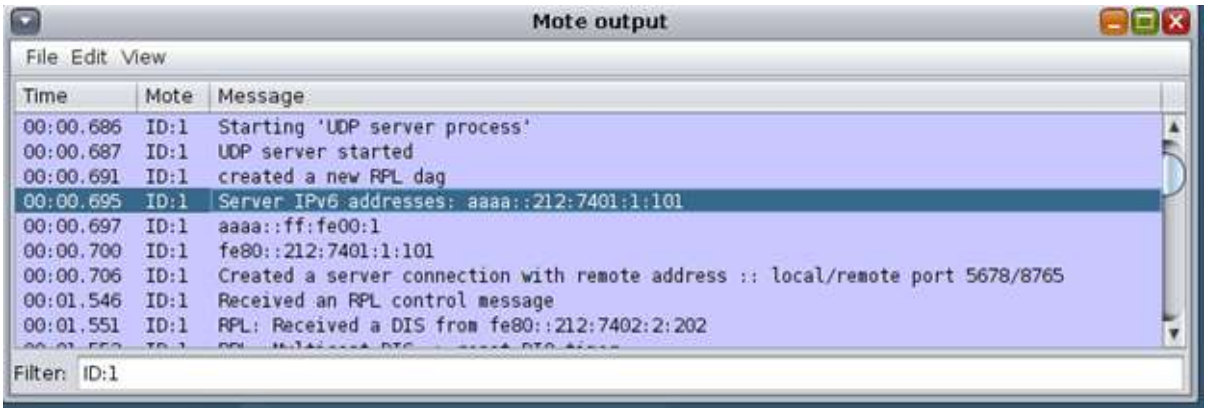

Fig 3.8 Server IPv6 address

As soon as the simulation gets started all nodes gets started at different times due to the random seed and all nodes are allotted IPv6 addresses. We will be using here ETX as objective function and hence ranks of nodes will be computed using ETX values as link metric.

After all nodes are allotted IPv6 addresses, the server node i.e. node 1 is set as root of RPL DODAG. The nodes start joining the RPL DODAG. Different nodes start sending RPL DIS message requesting the network information to join DODAG as shown in Fig. 3.9

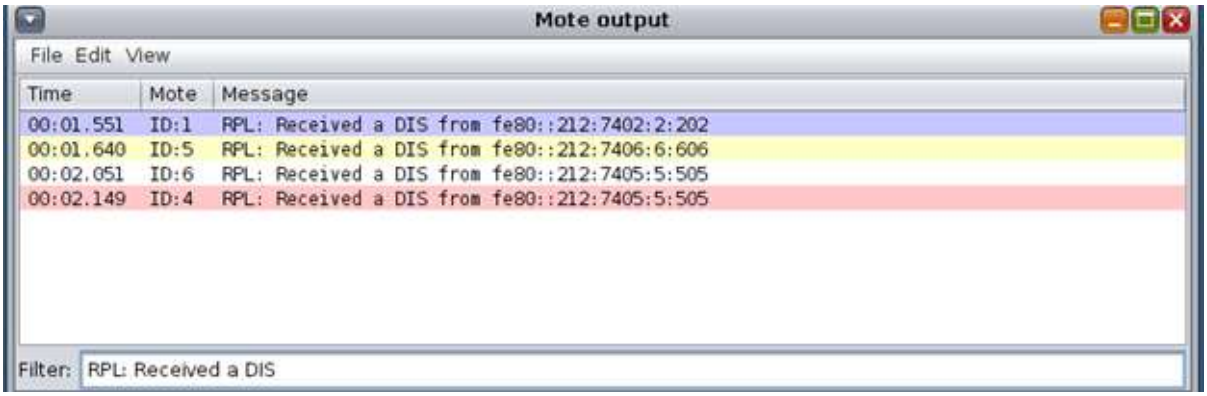

Fig 3.9 Nodes receiving DIS

Then in response to the DIS messages, nodes start sending DIO messages carrying network information as explained previously.

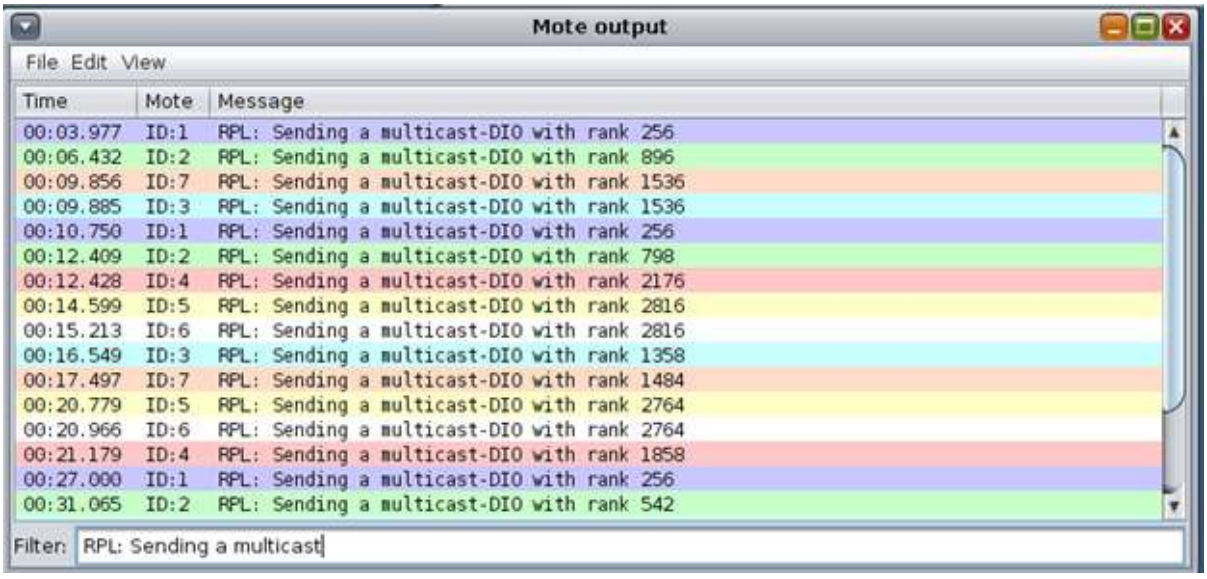

Fig 3.10 Nodes Sending DIO

Nodes multicast the DIO messages with the initial allotted ranks (using ETX as link metric). ALL nodes on the link receive the DIO messages and process them and add the sender to its neighbor cache and select its parent

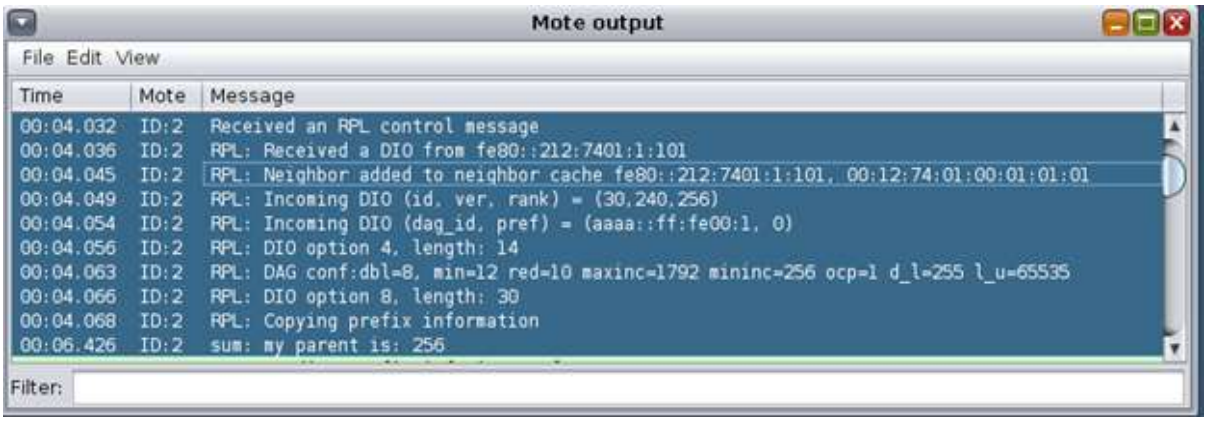

Fig 3.11 Processing DIO Message

After the simulation runs for specific time all nodes are assigned their parents and hence the RPL network gets formed. However the network formed is not permanent. Path metrics gets changed according to the formula shown previously (due to change in ETX values), due to which the rank of nodes gets changed. DIO messages with changed ranks are multicast on the respective links. Each time the node receives the DIO message the node process it and considers it as candidate parent. It then compares its path metric with the actual parent and then accordingly change its parent.

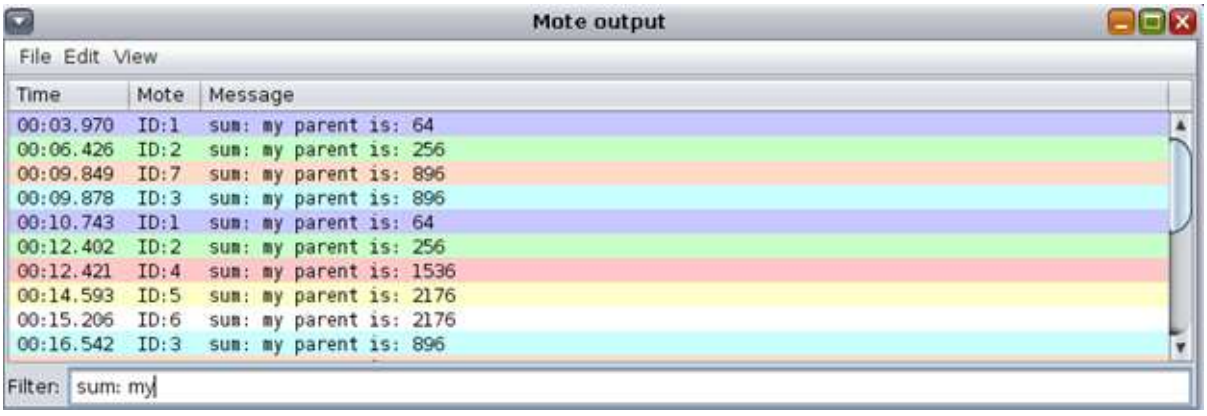

Fig 3.12 Nodes indicating their parents

The following Fig 3.13 shows the final status of all nodes. It shows the status of DAO messages being sent to their nodes

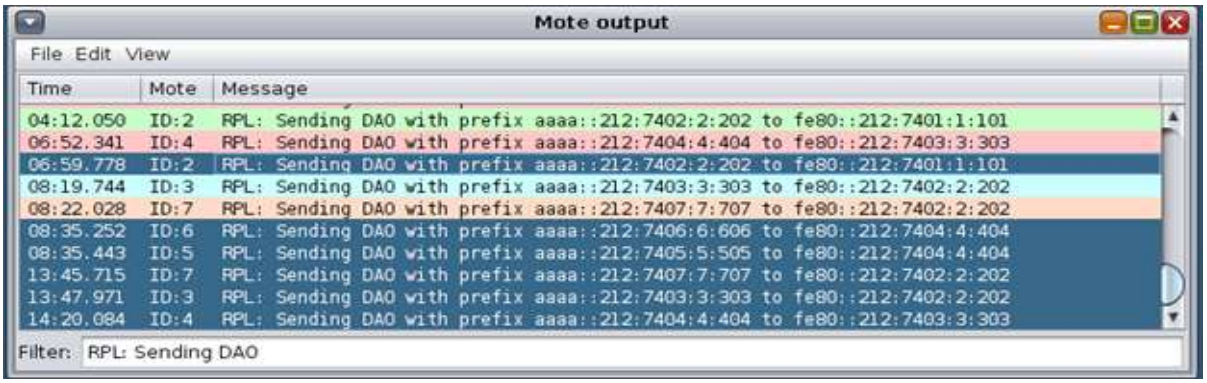

Fig 3.13 Final Status of all nodes

3.5 Proposed Methodology

RPL makes use of ranks for route discovery. Rank basically tells about the position of node in graph relative to each other and ensures that no cycles are present in the graph. However ranks in RPL are determined as per the definition of objective function that utilizes some routing metrics such as ETX (Expected Transmission Count) [7]. In our work we tend to modify the objective function to change ETX values to implement MITM.

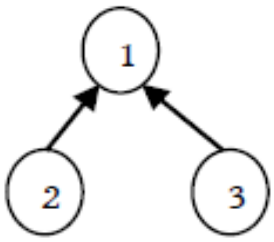

Fig 3.14 Normal topology

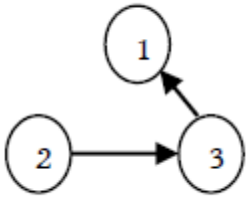

Fig 3.15 MITM Attack

To perform attack we can make play with objective function in following ways:

- By making the rank of attacker (node 3) node lower than parent node (node 1) so that its neighbour (node 2) selects attacker as its parent
- Make path cost from victim node to attacker node as zero

Chapter-4 Performance Analysis

The Experimentation has been performed on 3 types of topologies mainly:

- Random Topology
- Chain Topology
- Mesh Topology

4.1 Random Topology

The random topology consists of 29 client nodes and 1 server node randomly placed. Simulations are created in similar fashion as indicated in chapter 3. Fig 4.1 depicts the random topology.

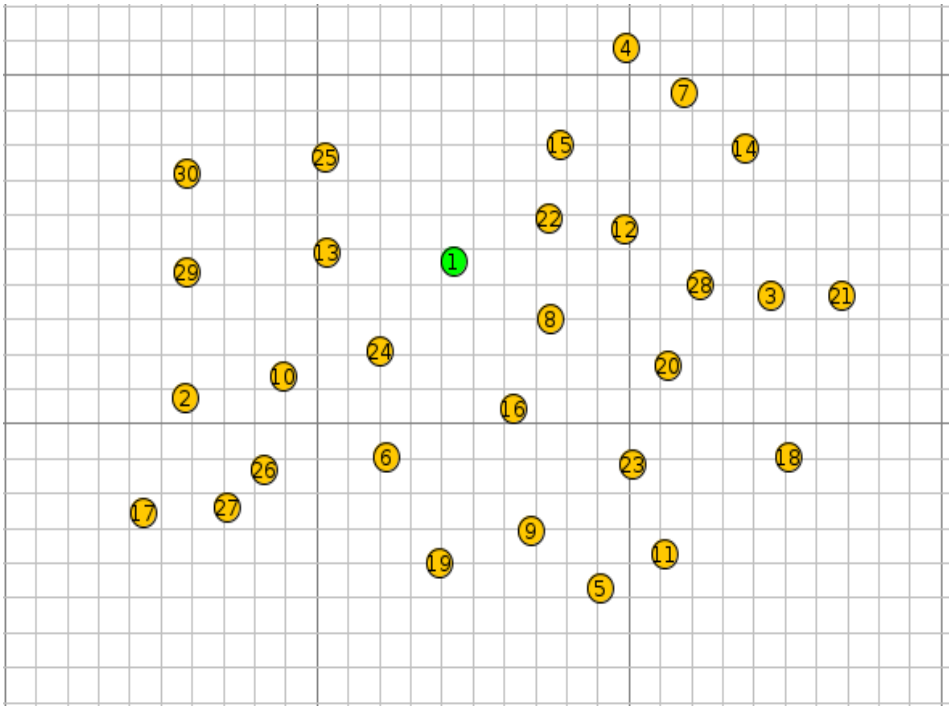

Fig 4.1 Random Topology

Simulations are carried out for 1800s and attack gets launched after 300s form the start of simulation. Fig. 4.2 depicts the network graph formed and provides information about the parent of the nodes.

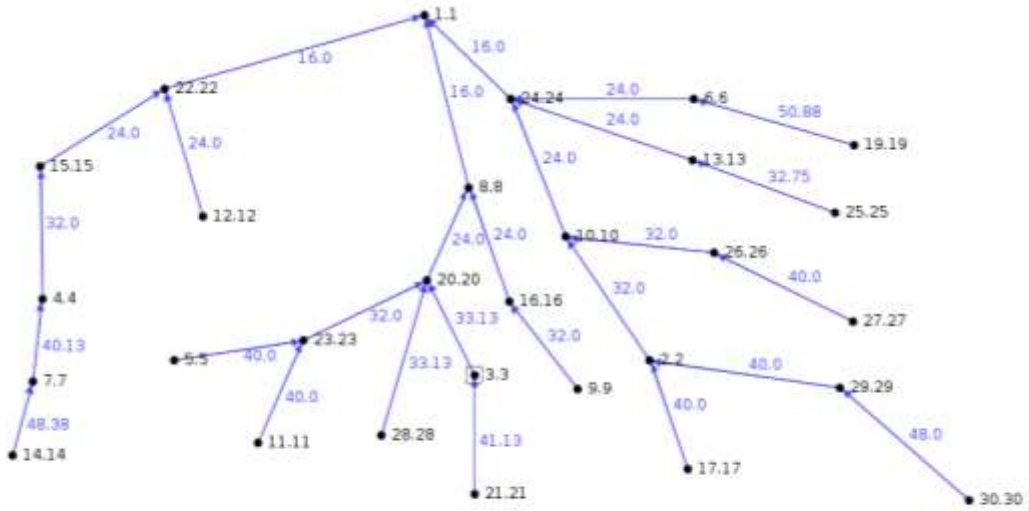

Fig. 4.2 Network Graph without Attack

The experimentation has been performed by gradually increasing the attacker nodes in the network and analysing their effect over the network. However, the attackers are chosen randomly.The table 4.1 provides the information about attacker nodes when number of attackers gets increased.

Number of Attackers	1	2	3
Attacker Node	Node 12	Node 12, Node 28	Node 12, Node 28, Node 13

Table 4.1 Position of Attacker corresponding to number of Attackers

The Graph in Fig 4.3 depicts node 12 being acting as an attacker node and node 15 and subsequent subgraph now sending its entire traffic to node 12 before sending to node 22. Node 12 is acting as a man in the middle for node 15 and node 22.

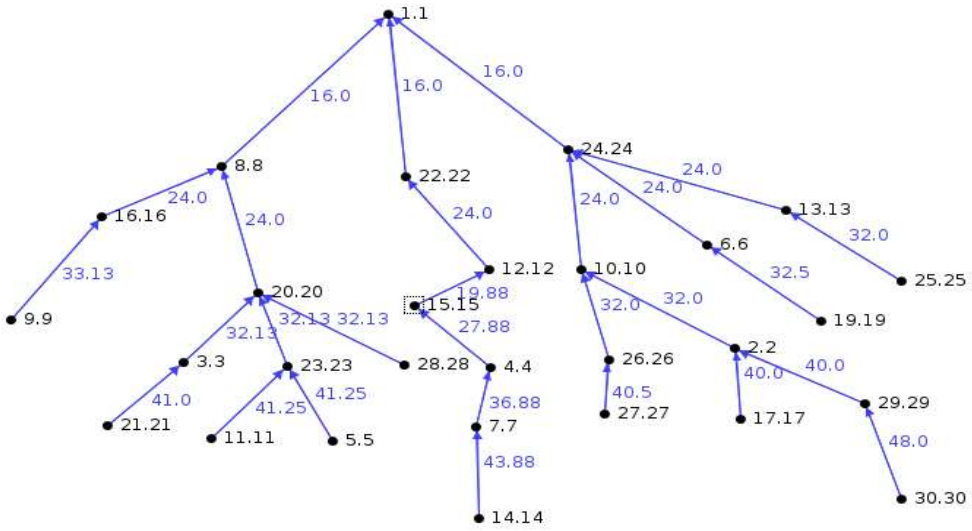

Fig. 4.3 Network Graph with one Attacker node

The graph in Fig 4.4 depicts the network graph being formed when number of Attacker nodes becomes two. In this case Node 12 and Node 28 are acting as attacker nodes.

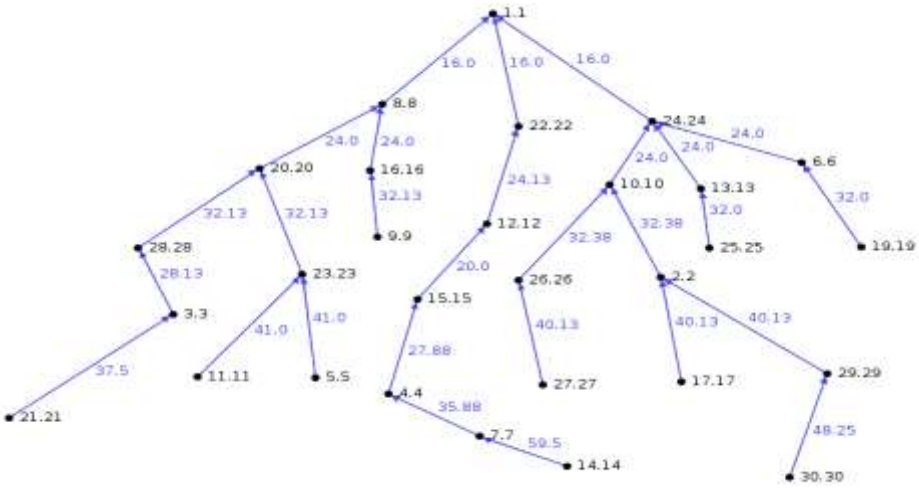

Fig 4.4 Network Graph with two attacker Nodes

The network graph in Fig 4.5 depicts the network graph being formed when number of Attacker nodes becomes 3. In this case Node 12, Node 28 and Node 13 are acting as attacker nodes. However in this case node 10 and subsequent subgraph is sending its traffic to node 13 before sending to node 24. Node 13 is acting as man in the middle besides nodes 28 and Node 12.

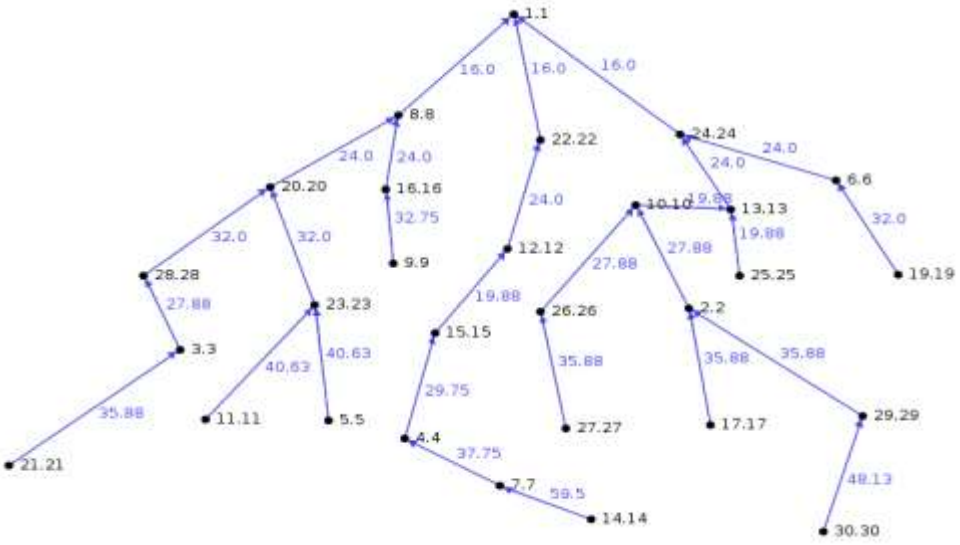

Fig 4.5 Network Graph with 3 Attacker Nodes

4.1.1 Analysis

Packet Delivery Ratio (PDR)

Packet Delivery Ratio is defined as the ratio of total packets received and total packets sent from all nodes to the sink.

Avg PDR = (Total Packets Received/Total Packets Sent)*100

Without Attack	Number of Attackers 1	Number of Attackers 2	Number of Attackers 3
99.76%	99.63%	99.13%	99.10%

Table 4.2 PDR wrt frequency of Attackers

Attack has no effect on Packet Delivery Ratio as expected as the attacker node is only intercepting the messages, not dropping them.

Packet Latency

Packet Latency is defined as time taken by a packet to reach sink node from sender node. We generally take average of latency of all packets form all nodes.

Total Latency = ∑ (Received Time – Sent Time)

Without Attack	Number of Attackers 1	Number of Attackers 2	Number of Attackers 3
57 sec	58 sec	58 sec	58 sec

Table 4.3 Latency corresponding to number of Attackers

Attack shows no impact on Packet Latency. Attacker node is not introducing any delay in the messages, hence making the attack more secluded.

Control Traffic Overhead

RPL makes use control messages to build and maintain the topology. This includes DIS, DIO and DAO messages.

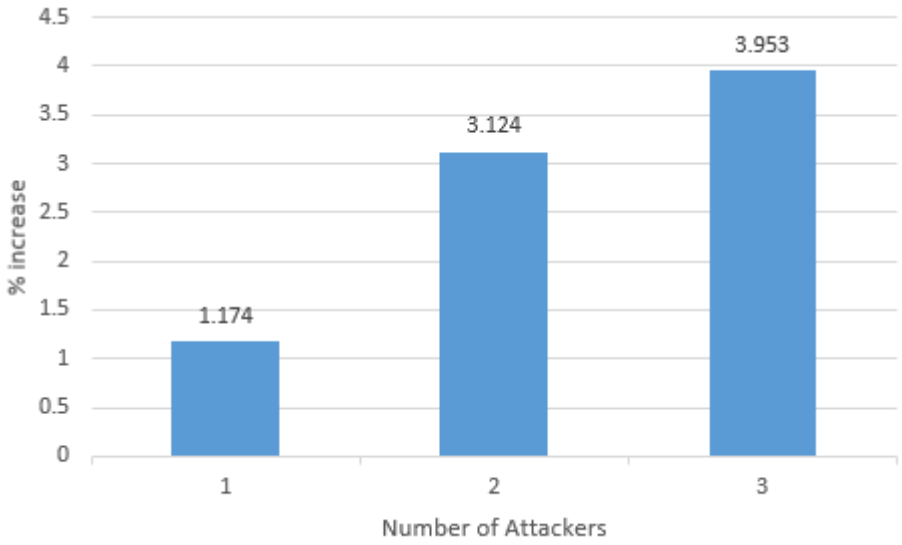

Fig 4.6 Percent increase in Control Messages

According to the specifications defined in [3], RPL sends control messages set according to the trickle timers in order to reduce the traffic over the network. However, when the DODAG is inconsistent the frequency of control messages gets increased [4]. In the experiment conducted, the attacker nodes located at different locations in the topology launches attack

causing fickle in the existing DODAG, due to which number of control messages gets increased till the DODAG gets stabilized.

Energy Consumption

Energy is crucial and expensive factor in an IoT network. Node 12 is the attacker node, forcing node 15 to become its child, due to which the whole subgraph shown grey in fig4.7 gets connected to node 12 hence modifying the entire topology. The experimental analysis shows increase in energy consumption of the nodes in entire subgraph from 4.919mw in normal scenario to 5.094mw under attack showing 3.55% increase. The change in topology during attack leads to more transmit and listen duty cycle forcing nodes to stay awake for longer durations and hence more consumption of energy

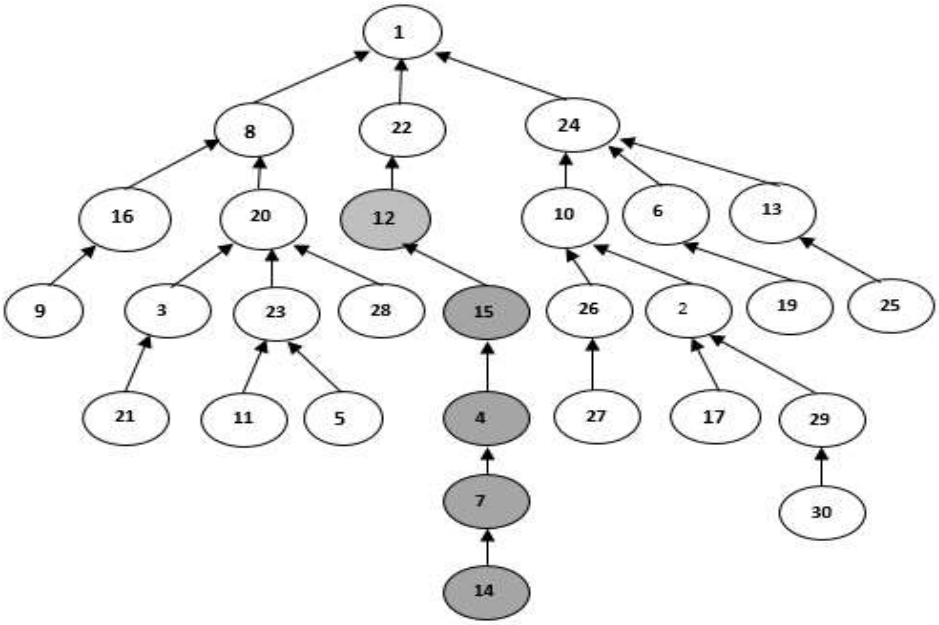

Fig. 4.7 Network Graph

4.2 Chain Topology

Chain Topology consist of 11 client nodes and 1 server node arranged in a linear fashion. Each node has only single option for parent. Fig 4.8 depicts the chain topology.

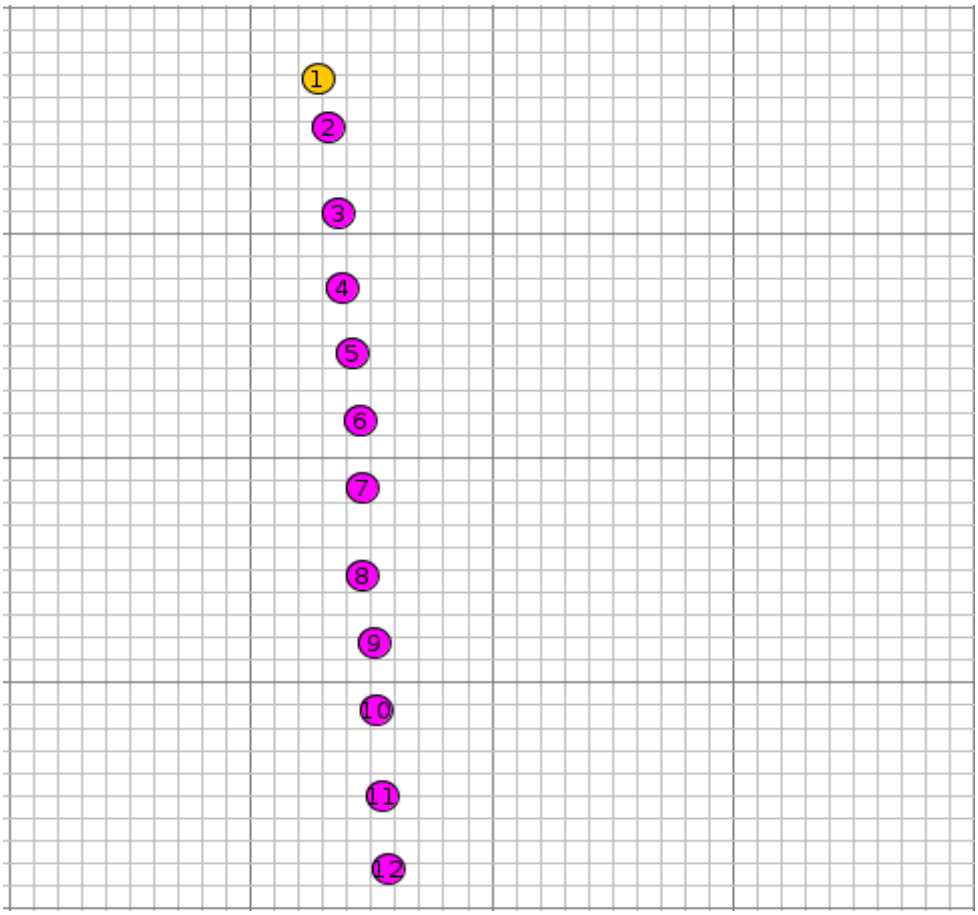

Fig. 4.8 Chain Topology

Simulations are carried out for 2700s and attack gets launched after 600s form the start of simulation. Fig. 4.9 depicts the network graph formed and provides information about the parent of the nodes

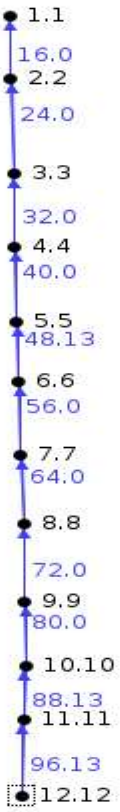

Fig 4.9 Network Graph for without Attack

The experimentation has been performed by changing the position of the attacker node in the network and analysing the effect over the network. The attacker node is gradually moved towards the root.

Status	**Attacker Position**
Without Attack	-
First Attack	Node 10
Second Attack	Node 7
Third Attack	Node 3

Table 4.4 Position of Attacker node

4.2.1 Analysis

Packet Latency

Without Attack	Number of Attackers 1	Number of Attackers 1	Number of Attackers 3
58 sec	58 sec	58 sec	58 sec

Table 4.5 Latency of Packets wrt attacker's frequency

Attack shows no impact on Packet Latency as expected.

Packet Delivery Ratio (PDR)

No changes have been observed in Packet Delivery Ratio

Control Traffic Overhead

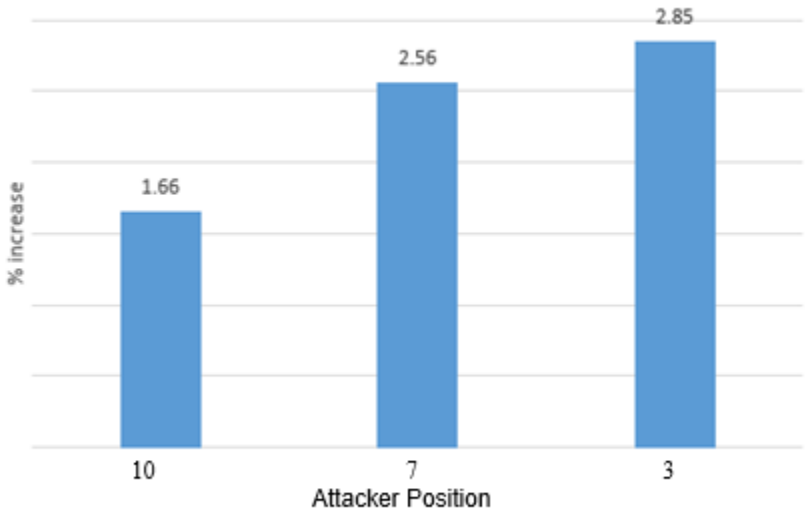

Fig. 4.10 % increase in Control Messages

Increasing trend in RPL control messages is observed as attacker node move closer to root.

As the rank of attacker node changes it generates DIO messages to inform about it to its child nodes who in turn updates their rank. If the attacker node is near to the root, it means all the nodes succeeding will have to update their ranks.

Energy Consumption

Total average energy of network is calculated

Avg. Energy = ∑ (Energy of all nodes)/(Total number of nodes)

Without Attack	Attacker node 10	Attackers node 7	Attacker node 3
1.081mw	1.087mw	1.091 mw	1 .094mw

Table 4.6 Avg. Energy Consumption of Network

4.3 Mesh Topology

Mesh Topology consists of 29 client nodes and 1 server node arranged in a way where each node is in range of every other node. Each node has maximum possible options for parent. Fig 4.11 depicts the mesh topology.

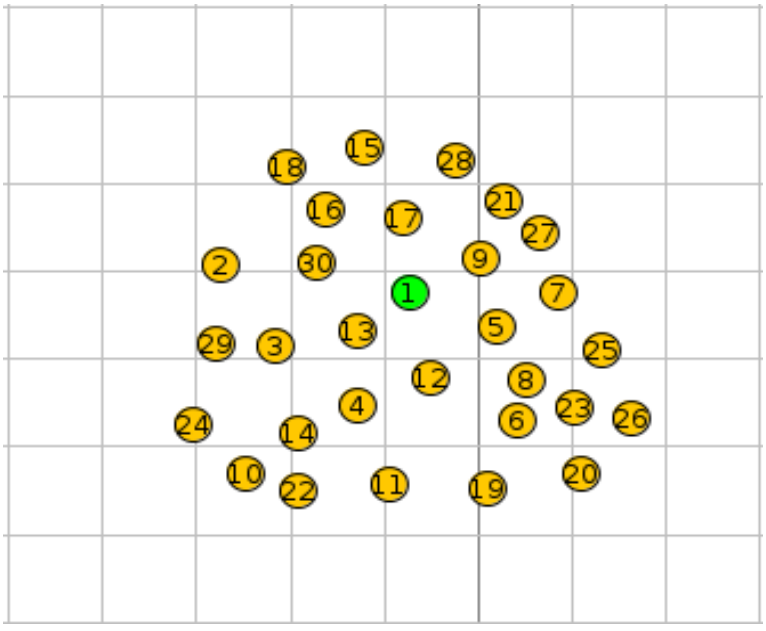

Fig. 4.11 Mesh Topology

Simulations are carried out for 2700s and attack gets launched after 600s form the start of simulation. Fig. 4.12 depicts the network graph formed and provides information about the parent of the nodes

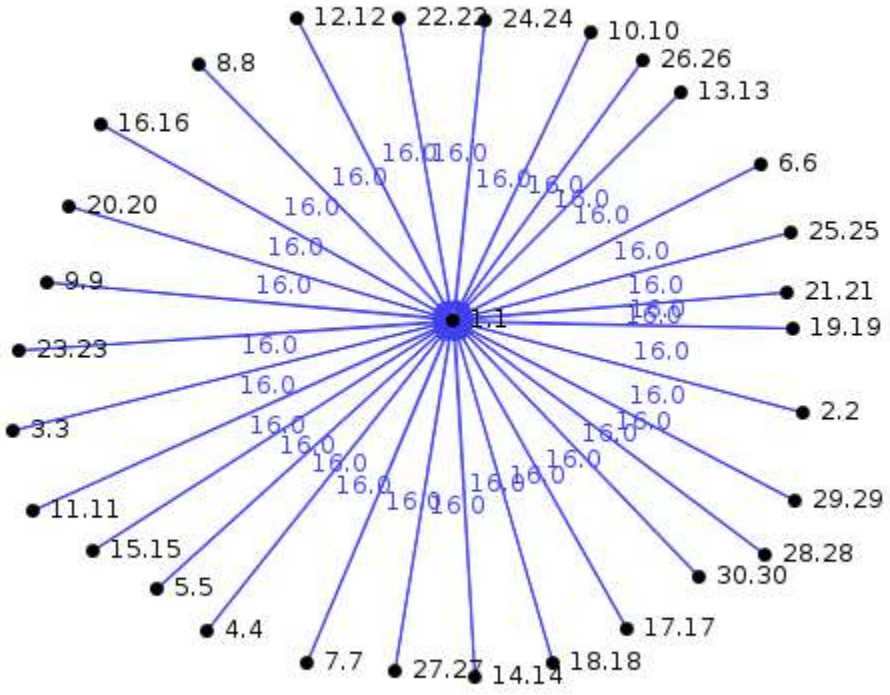

Fig. 4.12 Network Graph

4.3.1 Analysis

In case of mesh topology all nodes have maximum options of parents. However the experiment shows that when the simulation is performed without attack, all nodes tends to choose root as their parent because root has the minimum rank of 1. However in case of RPL it is not possible for the node to have rank less than root. [6] States that rank field in DIO messages that carry the rank of node is 16 bit unsigned integer field, which ensures that negative ranks are not allotted to the node. Hence this attack has no significance in case of mesh topology.

Chapter-5 CONCLUSION

5.1 Conclusion

IoT is a relatively new field and security in IoT is the main hindrance in the commercialization of the applications provided by IoT. In this project, we have proposed a Man in The Middle Attack in which a malicious node decreases its rank allowing victim nodes to select it as their forwarding node and all data of victim node passes through the Attacker node. The experimentations put forward the increased scope of man in the middle attack. It shows that the attack is not only limited to IPv4, but can also be implemented on relatively new RPL protocol for IoT. The attacker node preserves the nature of the attack and silently intercepts the messages without causing any change in the network.

This attack poses serious threat to the availability or authenticity of information. However, this attacks provides the option to the attacker to either manipulate the information or prevents the messages from being forwarded.

5.2 Future Work

There are several other aspects of security of IoT that need to be explored as a future study as follows.

1) Developing mitigation strategy for the attack proposed.
2) Considering the scope of other attacks in field of IoT.

References

[1] Tsvetkov, Tsvetko, and Alexander Klein, "RPL: IPv6 routing protocol for low power and lossy networks," *Network* 59 (2011).

[2] Xia, Feng, Laurence T. Yang, Lizhe Wang, and Alexey Vinel,"Internet of things," *International Journal of Communication Systems* 25, no. 9 (2012): 1101.

[3] Vasseur, J., Navneet Agarwal, Jonathan Hui, Zach Shelby, Paul Bertrand, and Cedric Chauvenet, "RPL: The IP routing protocol designed for low power and lossy networks," *Internet Protocol for Smart Objects (IPSO) Alliance* 36 (2011).

[4] https://www.ietf.org/rfc/rfc2460.txt

[5] https://tools.ietf.org/html/rfc826

[6] https://tools.ietf.org/html/rfc4443

[7] https://tools.ietf.org/html/rfc6550

[8] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized link state routing protocol for ad hoc networks,” in Multi Topic Conference, 2001, IEEE INMIC 2001, Technology for the 21st Century. Proceedings. IEEE International, 2001, pp. 62–68.

[9] C. E. Perkins and E. M. Royer, “Ad-hoc on-demand distance vector routing,” in Second IEEE Workshop on Mobile Computing Systems and Applications, 1999, Proceedings, WMCSA ’99, 1999, pp. 90–100.

[10] V. D. Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in INFOCOM ’97, Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings IEEE, 1997, vol. 3, pp. 1405 –1413.

[11] J. P. Vasseur, R. Kelsey, R. Struik, P. Levis, T. Winter, A. Brandt, J. Hui, K. Pister, T. Clausen, and P. Thubert, “RPL: IPv6 Routing Protocol for Low power and Lossy Networks,” [Online]. Available: http://tools.ietf.org/html/draft-ietf-roll-rpl-19. [Accessed: 02-Nov- 2011].

[12] A. Dunkels, “Full TCP/IP for 8-bit architectures,” in Proceedings of the 1st international conference on Mobile systems, applications and services, New York, NY, USA, 2003, pp. 85–98.

[13] N. B. Priyantha, A. Kansal, M. Goraczko, and F. Zhao, “Tiny web services: design and implementation of interoperable and evolvable sensor networks,” in Proceedings of the 6th ACM conference on Embedded network sensor systems, New York, NY, USA, 2008, pp. 253–266.

[14] A. Christian and J. Healey, “Gathering motion data using featherweight sensors and tcp/ip over 802.15. 4,” in Workshop on On-Body Sensing, Osaka, Japan, 2005.

[15] M. Durvy, J. Abeillé, P. Wetterwald, C. O’Flynn, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne, and A. Dunkels, “Making sensor networks IPv6 ready,” in Proceedings of the 6th ACM conference on Embedded network sensor systems, New York, NY, USA, 2008, pp. 421–422.

[16] J.-P. Vasseur and A. Dunkels, Interconnecting Smart Objects with IP: The Next Internet. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010.

[17] N. Dejean, J. P. Vasseur, D. Barthel, M. Kim, and K. Pister, “Routing Metrics used for Path Calculation in Low Power and Lossy Networks.” [Online]. Available: http://tools.ietf.org/html/draft-ietf-roll-routing-metrics-19. [Accessed: 04-Jan-2012].

[18] Maynard, Peter, Kieran McLaughlin, and Berthold Haberler. "Towards understanding man-in-the-middle attacks on iec 60870-5-104 scada networks." In *Proceedings of the 2nd International*

[19] https://tools.ietf.org/html/rfc4861

[20] Nikander, Pekka, James Kempf, and Erik Nordmark,” *IPv6 neighbor discovery (ND) trust models and threats*,” No. RFC 3756. 2004.

[21] Mayzaud, Anthéa, Rémi Badonnel, and Isabelle Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things," *International Journal of Network Security* 18, no. 3 (2016), pp.459-473.