



AMERICAN UNIVERSITY OF BEIRUT

TRUST AWARE ROUTING PROTOCOL FOR LOW POWER AND  
LOSSY NETWORK

by  
HASSAN SOUHIEL KHALIL

A thesis  
submitted in partial fulfillment of the requirements  
for the degree of Master of Science  
to the Department of Computer Science  
of the Faculty of Arts and Sciences  
at the American University of Beirut

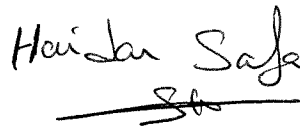
Beirut, Lebanon  
April-2019

AMERICAN UNIVERSITY OF BEIRUT

TRUST AWARE ROUTING PROTOCOL FOR LOW POWER AND  
LOSSY NETWORK

by  
HASSAN SOUHIEL KHALIL

Approved by:




---

Dr. Haidar Safa, Professor  
Computer Science

Advisor

---


Dr. Mohamed Nassar, Associate Professor  
Computer Science



Member of Committee

---

Dr. Wassim El Hajj, Associate Professor  
Computer Science



Member of Committee

Date of thesis defense: April-12-2019

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Khalil Hassan Souhail  
Last First Middle

Master's Thesis                       Master's Project                       Doctoral Dissertation

I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One --- year from the date of submission of my thesis, dissertation, or project.**  
**Two --- years from the date of submission of my thesis, dissertation, or project.**  
**Three --- years from the date of submission of my thesis, dissertation, or project.**

[Signature]                      May - 6 - 2019  
Signature                      Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

## ACKNOWLEDGMENTS

I am very thankful to almighty Allah for answering my prayers and giving me the strength and patience to plod along the whole way up to the end during this research work.

This thesis is the culmination of my journey of master's degree in computer science which was just like climbing a peak gratefully and carefully accompanied with encouragement, trust and hardship. When I found myself at top experiencing the feeling of fulfillment, I realized though only my name appears on the cover of this dissertation.

This dissertation would not have been finished properly without the support and motivation of several people who, in different ways, contributed in the completion of this study.

I would like to express my sincere gratitude to my advisor Dr. Haidar Safa for the continuous support of my master study and related research, for his patience, motivation, and great knowledge and for his guidance to help me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Wassim El-haj and Dr. Mohammad Nassar, for their valuable comments and encouragement, also for the hard questions which incited me to widen my research from various perspectives.

To my parents, sisters and brothers, for your love and support, I want to thank you for all the years you spent in loving me and giving me the strength to reach for the stars and pursuit my dreams.

## AN ABSTRACT OF THE THESIS OF

Hassan Souhiel Khalil

for

Master of Science

Major: Computer Science

Title: Trust Aware Routing Protocol for Low Power and Lossy Network

RPL, routing protocol for low power and lossy network is considered as the de-facto routing protocol for the internet of things (IoT). IoT is characterized by diverse devices that interconnect with each other on a cooperative basis in order to achieve certain objectives. IoT devices are constrained in terms of memory usage, processing and power consumption. Moreover, Security in IoT networks is a challenging task that is hard to achieve because the RPL routing protocol is subject to several internal and external attacks such as dropping messages, misleading the requesting nodes, behaving inaccurately, etc. Therefore, detecting and isolating these malicious nodes leads to more reliable and secure network. In this thesis, we propose a trust aware routing protocol for low power and lossy network by adding new trustworthiness metrics during the construction and maintenance of RPL network topology. Each node updates dynamically the trust metrics of its neighbors based on its observations and its neighbors' recommendations. We have evaluated the performance of the proposed approach through simulations. Obtained results showed that the proposed approach outperformed the existing ones when measuring parameters such as remaining energy in the network, delay, throughput, and percentage of malicious nodes detected while securing and protecting the network from attacks.

# CONTENTS

ACKNOWLEDGMENTS..... iv

ABSTRACT ..... v

Chapter

**1. INTRODUCTION ..... 1**

1.1. Background..... 1

1.2. Motivation..... 2

1.3. Problem Statement..... 3

1.4. Objectives and Contribution..... 3

1.5. Thesis Plan..... 4

**2. UNSECURED ROUTING IN INTERNET OF THINGS: ..... 5**

2.1. Internet of Things: History and Application Areas..... 5

2.2. IoT Topology ..... 7

2.2.1. 6LoWPAN Overview ..... 7

2.2.2. 6LoWPAN System Stack ..... 10

2.2.3. Routing in IoT and RPL ..... 11

2.2.4. RPL Overview:..... 12

2.2.5. DODAG Construction..... 15

2.2.6. Metrics, Constraints, and Objective Functions: ..... 20

2.2.7. Loop Detection and Avoidance:..... 22

2.2.8. Trickle Timer: ..... 22

2.3. Security in IoT: ..... 22

**3. APPROACHES TO MITIGATE SOME ATTACKS IN RPL 29**

3.1. Towards a trust computing architecture for RPL in CPS [14]..... 29

3.2. Trust-based service management for social IoT systems [15]..... 31

3.3.Hierarchical trust management for wireless sensor networks and its application to trust-based routing [16] .....	35
3.4.Trust-based RPL for the Internet of Things [17] .....	38
3.5.Design of primary and composite routing metrics for RPL [18] .....	40
3.6.Using trust management to defend against routing disruption attacks [19].....	41
3.7.Link Reliable and Trust Aware RPL (LT-RPL) for IoT .....	42
3.8.Summary .....	46
<b>4. PROPOSED TRUST AWARE RPL .....</b>	<b>48</b>
4.1.Trust Model Basic Concepts .....	48
4.2.Proposed Trust-Aware Routing protocol for Low Power and Lossy Network (TARPL-LLN) .....	49
4.2.1. Network Construction .....	49
4.2.2. First Level: Social trust Relationship (filtering step) .....	51
4.2.3. Second Level: QoS Level (preferred parent selection) .....	52
4.3.Interaction of 2 Neighbor Nodes.....	54
4.4.Trust representation in RPL DIO message: .....	55
4.5.Illustrated Examples: .....	57
4.5.1. Self-promoting attack .....	57
4.5.2. White-Washing attack .....	57
4.5.3. Bad-Mouthing Attack.....	58
4.5.4. Ballot-Stuffing Attack.....	58
4.5.5. Sinkhole Attack.....	59
4.5.6. Hello Flooding Attack.....	59
4.5.7. Rank Attack.....	59
4.5.8. Selective Forwarding Attack.....	59
4.5.9. Blackhole Attack.....	60
4.6.How TARPL-LLN differs from other works.....	60
<b>5. PERFORMANCE EVALUATION.....</b>	<b>62</b>
5.1.The Contiki Simulator.....	62
5.2.Simulation, Metrics and Parameters .....	63



5.2.1. Phases of Evaluation and Network Setup: .....	64
5.2.1.1. Failing in Transmission.....	65
5.2.1.2. Network Setup.....	66
5.2.1.3. Calculating the performance metrics .....	68
5.3. Required Numbers of Runs.....	69
5.4. Phase 1: Performance of OF0, MRHOF and TARPL-LLN with no attacks .....	70
5.4.1. Remaining Energy in the Network.....	71
5.4.2. Packet Delivery Ratio.....	73
5.4.3. Latency .....	73
5.4.4. Phase 1 outcomes .....	74
5.5. Phase 2: Performance of MRHOF and our Approach with the Presence of Malicious Nodes .....	74
5.5.1. Performance with 20% Malicious Nodes.....	75
5.5.2. Packet Delivery Ratio.....	78
5.5.3. Packet Latency .....	79
5.5.4. Throughput .....	80
5.5.5. Packet Loss Ratio .....	81
5.6. Comparing TARPL-LLN with 2 other approaches .....	82
5.6.1. Remaining Energy.....	82
5.6.2. Packet Latency .....	83
5.6.3. Percentage of Malicious Detection .....	84
5.7. Overhead on The Network.....	85
5.8. Effect of $\alpha$ and $\beta$ on trust Evaluation .....	86
5.8.1. Effect of $\alpha$ on trust evaluation.....	87
5.8.2. Effect of $\beta$ on trust evaluation.....	88
<b>6. CONCLUSION .....</b>	<b>90</b>
<b>7. REFERENCES: .....</b>	<b>91</b>

*To my parents,  
for their unconditional love and endless patience*

# CHAPTER 1

## INTRODUCTION

Popular demand on internet nodes combined with technology advances is driving extensive diffusion of an Internet of Things (IoT) that could, like the present Internet, participate valuably to economic development and military ability as reported in 2008 by the U.S. National Intelligence Council (NIC) [30]. The widespread of this technology arises many challenges such as scalability, connectivity, privacy, security, etc. Security is one of the most important requirements for IoT, therefore, in this thesis we aim to build a trust aware routing protocol for IoT in order to detect and avoid variety of attacks. In this chapter, we present an overview of IoT networks, the motivation behind our work, problem definition, objectives and contribution, and finally thesis plan.

### **1.1. Background**

Smart, low processing, and low power things can interact, interconnect, collaborate, and transfer sensing information to the internet using heterogeneous wireless technologies without the intervention of human. This concept is known as the internet of things (IoT) networks. IoT is going to be the Internet of the future in which a large number of devices will communicate with each other to make our society cleverer. This technology is expected to change our world as the Internet did. To overcome the connectivity issue of such networks, the Internet engineering task force (IETF) designed an appropriate solution under the name 6LOWPAN (IPv6 over low power and lossy network). 6LOWPAN is a networking adaptation layer that allows IPv6 packets to be carried within small link layer frames, such as those defined by IEEE 802.15.4 or WIFI [5]. IoT network is connected to the Internet through an access point (AP) called 6LOWPAN border

router (6LBR). In order to achieve this mechanism, IETF formed a required wireless communication protocol stack for the IoT that meets the important criteria of reliability, power-efficiency, Internet connectivity, and communications between constrained sensing devices and Internet devices outside of a local sensor network, thus laying the ground for the creation of new services and distributed applications. The main features of this stack are, physical and MAC layers, 6LOWPAN adaptation layer and the routing over 6LOWPAN that is supported by the routing protocol over low power and lossy network (RPL) as shown in the figure 1.1.

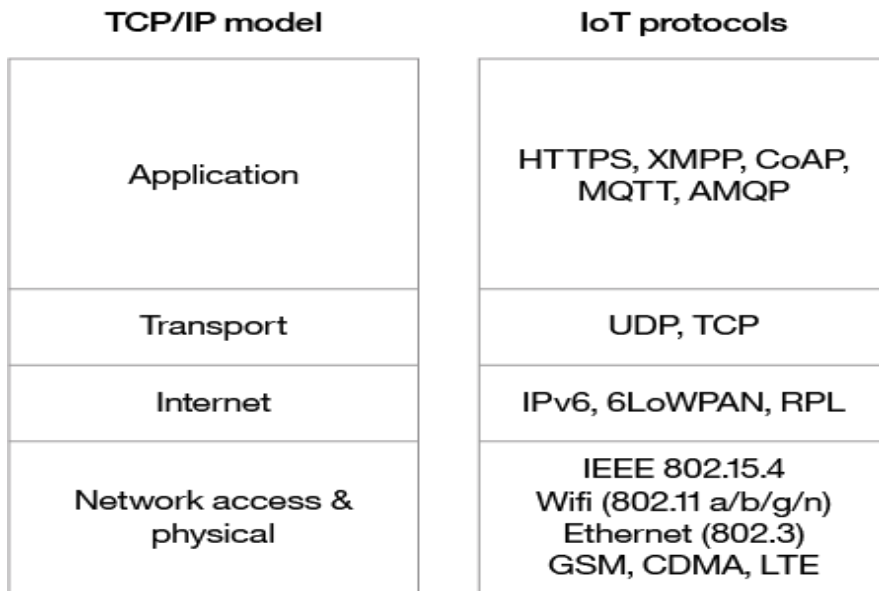


Figure 1.1: Protocol Stack for IoT

## 1.2. Motivation

The RPL protocol was recently standardized as a routing protocol for the IoT network. It works on processing and forwarding the packets while minimizing the consumption of energy, decreasing the communication delays, and satisfying many IoT constraints. RPL is more efficient than other routing protocols [9], since it is able to quickly build network topology, distribute routing knowledge among nodes, and implement some measures to efficiently use available resources such as energy [8] [31]. Even though, RPL supports secure messages that provide

confidentiality and integrity of data packets in transit and authentication between devices, an attacker can still launch a number of attacks against the IoT network. Examples of these attacks are selective forwarding attack, sinkhole attack, sybil attack, hello flooding attack, wormhole attack, blackhole attack. Moreover, RPL specifications do not contain any counter measures or self-healing capacity against most of these attacks [12]. As a result, security issue is a major bottleneck of the future development of IoT. hence, providing solutions to information security problems in IoT networks has become a significant research area.

### **1.3. Problem Statement**

In the literature, only few works were proposed on trust management for IoT. These proposed approaches do not provide solution to avoid many existing and popular attacks and don't take into consideration the fact that most IoT devices are constrained in terms of processing power, energy and storage. In this thesis, we incorporate the trust concept in IoT routing protocols to mitigate these attacks by isolating malicious nodes. This can be achieved by assigning each node a trust value according to its past performance in routing. Then, such trust values are used to help a node in choosing a secure and efficient route.

### **1.4. Objectives and Contribution**

In this thesis, we aim to propose a trust aware routing protocol for low power and lossy network that is able to detect and avoid malicious nodes from the network topology and classify nodes according to their trust and reputation level. When a malicious node drops, redirects messages or launches other attacks, its neighbor nodes can detect this malicious behavior and redirect their paths towards the route by eliminating these malicious nodes. The trust management system also makes sure that if any node provides incorrect/inaccurate feedback then its credibility is reduced and thus this node's evaluations should not affect others' reputation.

Our main contributions can be summarized as follows:

1. Surveying existing RPL trust management system and approaches.
2. Proposing a new trust aware routing protocol for low power and lossy network.
3. Evaluating the performance of the proposed approach and comparing it with the existing approaches in terms of mitigating attacks, preserving energy, increasing throughput and decreasing delay.

### **1.5. Thesis Plan**

The remainder of this thesis is organized as follows. In chapter 2, we present the basic concepts and background of IoT, its routing protocol, and the attacks that can be launched against them. Chapter 3 surveys existing approaches to detect attacks in IoT. In chapter 4 we describe our proposed trust aware routing protocol for low power and lossy network. Chapter 5 evaluates the performance analysis of our proposed model and analyze the obtained results. Finally, in chapter 6 we conclude our work and provide some limitations that can for the ground for future work.

## CHAPTER 2

### UNSECURED ROUTING IN INTERNET OF THINGS:

Internet of things (IoT) plays an important role in our daily life where it provides many tools that enhance the environment and make it more comfortable and adequate. In this chapter, we present the basic concept of IoT focusing on most relevant protocols and technologies. Then, we explain the routing in IoT identifying some existing vulnerabilities that can be exploited to launch attacks on IoT.

#### **2.1. Internet of Things: History and Application Areas**

The internet of the future is going to be the internet of things (IoT). A large number of devices will communicate with each other to make our society cleverer. These devices might be physical or virtual objects such as home appliance devices, social networks accounts, computers and many other objects embedded with electronics, software, sensors, actuators, etc. The initial concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first Internet-connected device that is able to report its inventory and state of products [1].

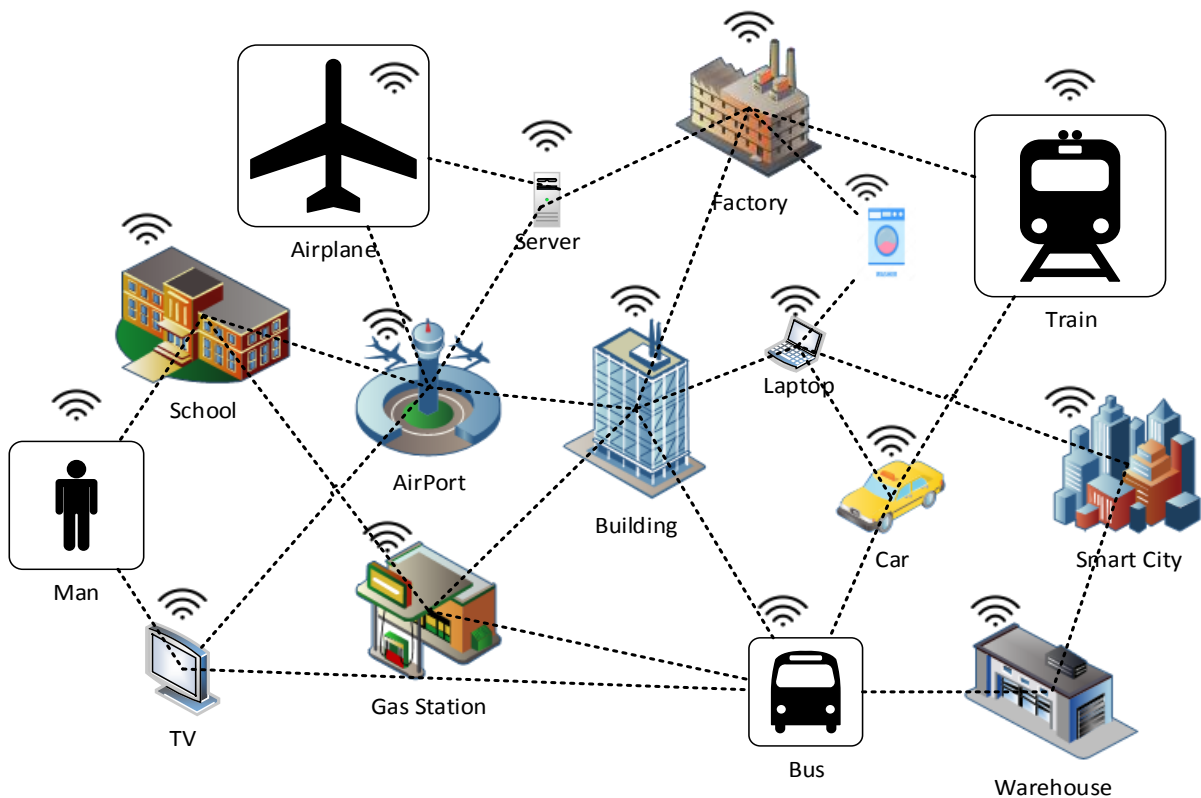
The combination of IoT technology in many vital domains in our lives was the focus of many recent researches. Indeed, currently IoT applications exist in almost every domain and will continue to play an important role in our future life. More specifically, figure 2.1 shows some usages of IoT in different domains such as:

1. In healthcare system, IoT offers tools ensuring patients are cared for better, healthcare costs reduced expressively, and treatment results are improved. In addition, using

intelligent application in this domain reduces the errors which enhances patients' experiences and the management of drugs [2].

2. In building and home automation, where IoT plays an essential role in realizing smart homes to make our lives easier, more convenient and comfortable [3].
3. In environmental monitoring, IoT sensors can take a highly labor-intensive process and make the environment simple and efficient.
4. In the infrastructure management, governments can use interconnected and intelligent devices to build systems for improving water distribution systems, reducing city traffic congestion, or making the electricity grid more efficient [4].

Moreover, IoT is being used in energy management, transportation systems and many other fields.



**Figure 2.1** The IoT technology could be included in many domains. Like smart homes, healthcare systems, energy management and transportation system.



Using things in all of these fields led to increase the number of interconnected devices in the internet, where according to the Federal Trade Commission (FTC), the number of IoT devices has already outnumbered the population and the number of IoT related wireless devices will be about 26 billion by 2020 [41]. However, connecting this large number of things through the Internet brings many challenges, such as

1. Scalability which is the ability of a system to handle a growing amount of works.
2. Resource constraints where the nodes are constrained in terms of memory, processor, and energy.
3. Mobility of IoT nodes in many application, interoperability, security and privacy.

Even though, these problems are being addressed and many solutions were proposed, they still constitute a wide area of research.

## **2.2. IoT Topology**

To integrate IoT applications to the internet, the internet engineering task force (IETF) workgroup (WG) designed an appropriate solution under the name 6LOWPAN (IPv6 over low power and lossy network) that is suitable to such type of network. Also, since large number of data will be generated by IoT applications that are built based on some constrained devices and lossy network, a compatible routing protocol was invented for generating an intelligent routing topology.

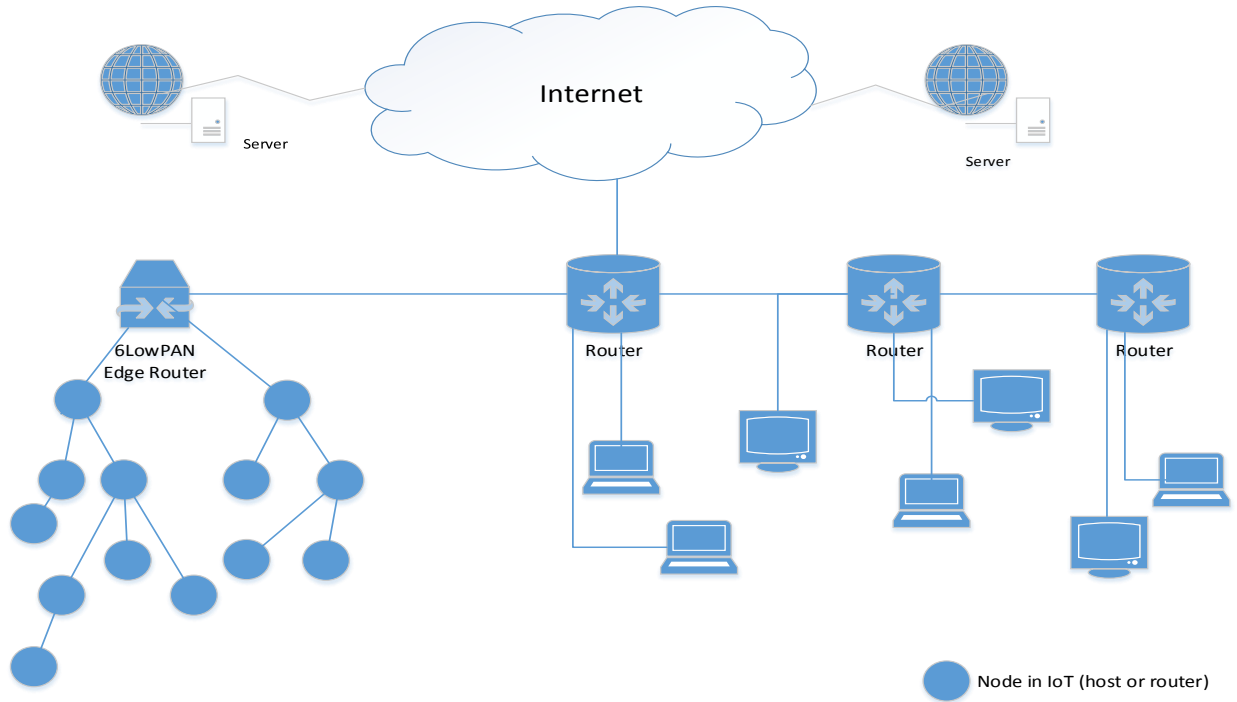
### **2.2.1. 6LoWPAN Overview**

6LOWPAN is a networking technology adaptation layer that allows IPv6 packets to be carried within small link layer frames, such as those defined by IEEE 802.15.4. Figure 2.2 shows

the design of an IPv6 based 6LOWPAN IoT network. Indeed, 6LOWPAN network is connected to the internet through an access point (AP) called edge router which handles three main actions:

1. The exchange of data between 6LOWPAN things and the internet.
2. The exchange of data between the devices inside 6LOWPAN network.
3. The generation and maintenance of the destination oriented directed acyclic graph (DODAG) [5] (explained later).

To achieve this mechanism, IETF defined a wireless communications protocol stack for the IoT to meet the important criteria of reliability, power-efficiency, Internet connectivity, and communications between constrained sensing devices and internet devices outside of a local sensor network, thus laying the ground for the creation of new services and distributed applications including both Internet and constrained sensing devices. 6LoWPAN incorporates IPv6-based infrastructures and IoT by allowing the IPv6 packets (they used IPv6 since the number of interconnected devices is so big) to be routed in a constrained network such as IEEE 802.15.4. Moreover, some protocols that support internet communication with sensing devices in the IoT are also needed. Hence, communication protocols were designed by IEEE and IETF enabling a standardized protocol stack, as shown in Figure 2.3(b). This stack permits the internet communications between devices even with the presence of constrained devices [6].

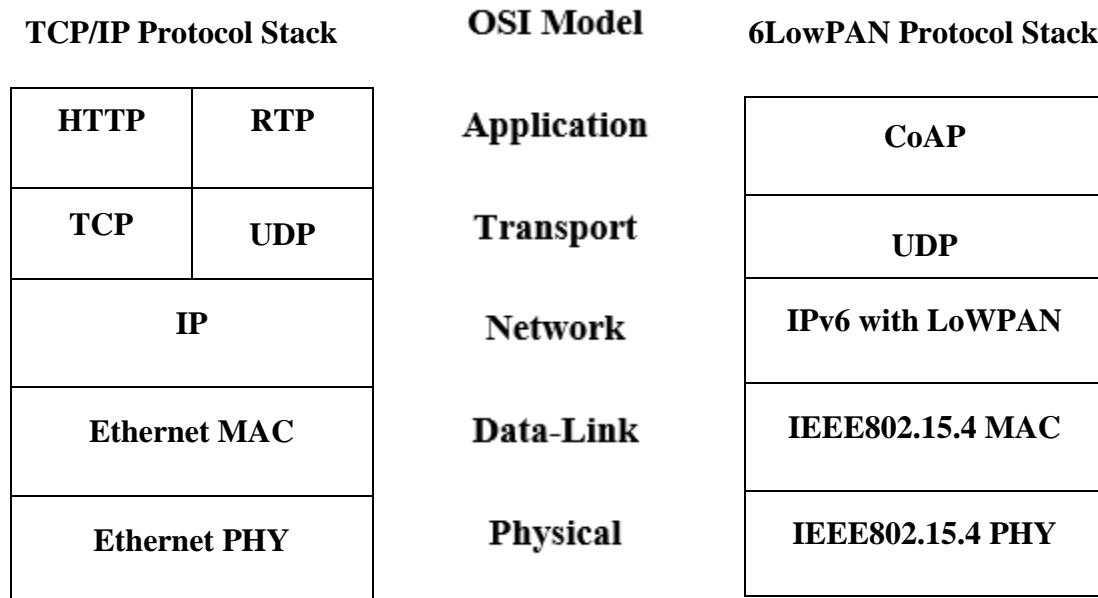


**Figure 2.2: This figure shows an example of an IPv6 network, including a 6LoWPAN network**

The main features of the 6LoWPAN protocol stack are listed below:

1. Low-energy communications at the physical (PHY) and Medium Access Control (MAC) layers are supported by IEEE 802.15.4 which sets communications rules at the lower layers of the stack.
2. Low-energy communication environments using IEEE 802.15.4 allow at most 102 bytes to be transmitted at higher layers of the stack. But this value is much less than 1280 bytes, the maximum transmission unit (MTU), required for IPv6. Hence, the 6LoWPAN adaptation layer handles this by enabling the transmission of IPv6 packets over IEEE 802.15.4, and providing mechanisms for packet fragmentation and reassembly, among other functionalities.

3. Routing over 6LoWPAN environments is supported by the routing protocol for low-power and lossy networks (RPL). RPL is considered as a de facto routing standard for IoT aiming to optimize the routing scheme for converge cast traffic pattern.
4. At the application layer, there is the constrained application protocol (CoAP) that supports communications. It is currently being designed at the IETF to provide interoperability in conformance with the representational state transfer architecture of the web.



**Figure 2.3: Protocol stacks that handle the interconnection and communication in the network.** (a) shows the protocol stack for TCP/IP network and (b) shows the 6LoWPAN stack that uses CoAP at the application layer and UDP at the transport layer and it has LoWPAN adaptation layer that allows the IPv6 packets to be transmitted through IEEE 802.15.4 physical layer.

### 2.2.2. 6LoWPAN System Stack

A complex application layer gateway was needed to make devices such as ZigBee, Bluetooth and proprietary systems connect to the internet. 6LoWPAN introduces an adaptation layer between link and network layers in the IP stack to allow the transmission of IPv6 datagrams over IEEE 802.15.4 radio link. All communications systems use a set of rules to format data and control the exchange. The most common model in data communication systems

is the Open Systems Interconnect (OSI) model that breaks the communication into seven fundamental layers (application, presentation, session, transport, network, data-link, and physical layers). Figure 2.3 shows a simplified OSI model (the first 3 layers are represented as application layer) with 2 typical stack examples used in TCP/IP devices (figure 2.3 a) and IoT devices (figure 2.3 b). The data link layer provides a reliable link between two directly connected nodes by detecting and correcting errors that may occur in the physical layer during transmission and receiving. 6LoWPAN adaptation layer provides a way to allow IPv6 packets to be carried within lossy network IEEE 802.15.4. The network layer routes data through the network over single or multiple hops. The transport layer generates communication sessions between applications running on end devices. Transmission control protocol (TCP) is the dominant transport protocol on the internet. However, TCP is a connection-based protocol with large overhead and therefore it is not suitable for devices demanding low power consumption. This is why, user datagram protocol (UDP) is used in IoT applications characterized by its lower energy consumption. Finally, the application layer is responsible for data formatting and insuring that the data is transported. The broadly used application layer on the internet is the HTTP protocol which is a text-based language with a large overhead. Hence, the industry and community have invented alternative application layer protocols, such as the constrained application protocol (CoAP) that is defined by IETF.

### **2.2.3. Routing in IoT and RPL**

Since IoT is being incorporated in large number of applications, enormous amount of data will be generated and consequently low power and lossy networks (LLNs) will be created. LLNs consist of constrained nodes with limited processing power and energy. These nodes are

interconnected by lossy links that are usually unstable with relatively low packet delivery rates. Therefore, an appropriate routing protocol must exist for generating an intelligent routing topology that can be further used to build a smart environment. Routing protocols use routing metrics to compute shortest paths that are needed by the nodes to send data to the sink node. Some routing protocols like IS-IS [26] and OSPF [27] use static link metrics that may reflect the bandwidth. In LLNs, the routing protocol requires the support of both static and dynamic metrics. Therefore, routing becomes more challenging for low-power and lossy radio-links, multi-hop mesh topologies, and frequently changed network topologies [7]. Existing MANET routing protocols such as AODV, DSR and OLSR were studied and analyzed in order to use them to invent an appropriate routing mechanism for the future IoT. However, Studies [8], [21], and [22] showed that these available routing protocols are not suited for LoWPAN networks for many reasons such as their consumption of energy, not handling failure cases to establish a connection and not taking into consideration nodes and links properties routes. At the end, the IETF ROLL working group preferred to design a new routing protocol for low power and lossy network (RPL) that works on processing and forwarding packets from routing optimization objectives by minimizing the consumption of energy, decreasing the communication delays and satisfying the IOT devices constraints. RPL is more efficient than other protocols, since it is able to quickly build network routes, distribute routing knowledge among nodes, and implement some measures to reduce the overall energy consumption.

#### **2.2.4. RPL Overview:**

RPL is a Distance Vector IPv6 routing protocol for LLNs arranges network topology as a destination oriented directed acyclic graph (DODAG), where each node has a specific DODAG

associated with it. DODAG is a directed graph, where all edges are oriented in such a way that no cycles exist, and it is routed at a single destination (i.e. at a single DAG root) [9]. RPL uses four key values for maintaining and identifying a topology as shown in Figure 2.4.

1. *RPLInstanceID* identifies an RPL instance that is a set of one or more DODAGs. IoT network may contain multiple *RPLInstanceIDs* that are optimized for different objective functions. All DODAGs in the same RPL instance use the same objective function (OF).
2. *DODAGID* is the identifier of a DODAG root. it is unique within the scope of an RPL Instance in LLN. Each DODAG is identified by a combination of *DODAGID* and *RPLInstanceID*.
3. *DODAGVersionNumber* is a counter that is increased by the root to form a new version of a DODAG. During network formation or during maintenance, the DODAGs may be reconstructed and new *DODAGVersionNumber* be derived from the old version by increasing it by one. The tuple (*RPLInstanceID*, *DODAGID*, *DODAGVersionNumber*) identifies the DODAG version.
4. Rank, it represents an abstract position of a node with respect to the DODAG route, where each node sets its rank according to the objective function. The rank computations maintain 2 main properties for any nodes M and N that are neighbors which are:
  - a. If  $Rank(M)$  is less than  $Rank(N)$ , the position of M is closer to the DODAG root than the position of N. Also, M may be a DODAG parent for Node N.
  - b. If  $Rank(M)$  equals  $Rank(N)$ , the position of M and N within the DODAG and with respect to the DODAG root are similar and identical. M cannot be a parent for N, since if this happens a loop will be formed.

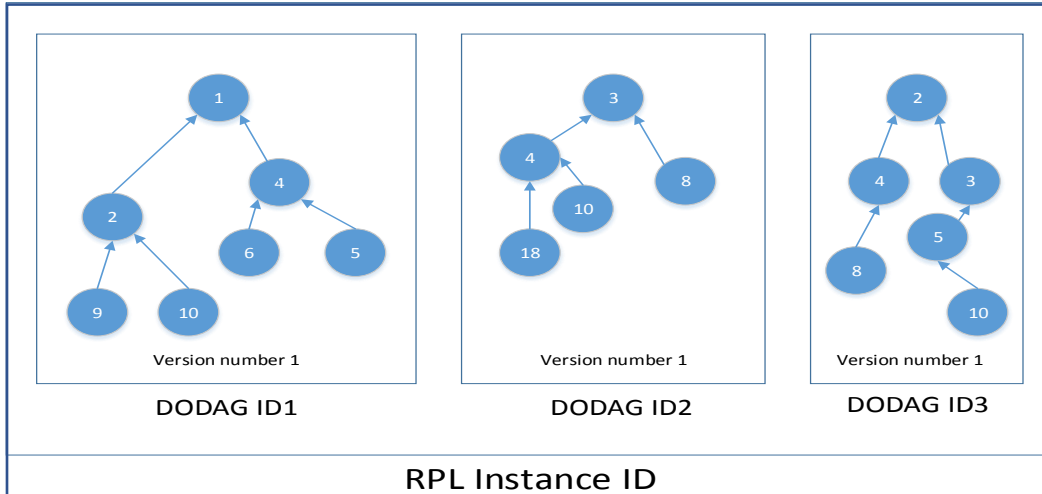


Figure 2.4: RPL Instance

RPL uses Objective functions (OFs) while selecting and optimizing the routes based on various constraints. Each instance is associated with a specialized objective function that helps nodes determine which DODAG it should join. It uses OF to translate key metrics and constraints into a rank and help node choose potential parents (i.e. set of parents that includes a preferred parent). Therefore, the main goal of OFs is to allow the node to join a DODAG version that offers good enough connectivity to a specific set of nodes. IETF proposed 2 main objective functions which are namely Minimum Rank with Hysteresis Objective Function (MRHOF) and Objective Function zero (OF0). OF0 is known as a basic OF that operates on parameters that are obtained from provisioning, the RPL DODAG configuration option and the RPL DODAG information object base container. OF0 is designed to find the nearest DODAG root that provides good connectivity with the DODAG root. Hence, OF0 uses a strategy that is analogous to finding the minimum hop-count between a node and the main root [10]. However, MRHOF is slightly more complicated and can compute a node's rank based on the additive metrics (such as node energy, hop-count, ETX, throughput, and latency) [24] [32]. RPL also contains 2 repair mechanisms which are the global repair and local repair, where in the former, the repair starts



from the DODAG root and it has the cost of additional control traffic in the network. In the latter, the repair takes place with the same DODAG version.

### 2.2.5. DODAG Construction

To build and maintain the DODAGs, RPL uses four types of control messages:

1. *DODAG information Object (DIO)* that is used to create the upward routing, from the nodes to the sink node, (i.e. multipoint to point communication).
2. *Destination Advertisement Object (DAO)* where it is used to create the path for downward routing, from the sink node which is the border router to other nodes (i.e. point to multipoint).
3. *DODAG Information Solicitation (DIS)* which is used to solicit or request a DIO from a RPL node, so any node wants to access the DODAG needs to broadcast a DIS control message to its neighbors.
4. *Destination Advertisement Object Acknowledgement (DAO-ACK)* that is sent as a unicast packet by a DAO recipient in response to a DAO message.

In addition to the previously discussed control messages, there is a consistency check (CC) message that is used for synchronization of counter values among communicating nodes and provide a basis for the protection against packet replay attacks.

Type	Code	Checksum
<b>Base</b>		
<b>Option(s)</b>		

**Figure 2.5: RPL Control Message**

The RPL control message is shown in Figure 2.5, where it consists of an ICMPv6 header followed by a message body. The message body is comprised of a message base and possibly several options. The RPL control message is an ICMPv6 information message with a type equals to 155. The code field recognizes the type of RPL control message, whether it is DIO, DAO, DIS, or DAO-ACK. The checksum is computed by the receiver for the purpose of detecting errors which may have been introduced during its transmission or storage.

Since most applications in IoT require upward routing, DIO messages play the main role in constructing such application. Figure 2.6 illustrates the format of a DIO message. The first field represents the *RPLInstanceID* while the second and the third fields are the sender's *DODAG version* and the rank of the message. the 'G' flag defines if the DODAG is grounded or floated (grounded DODAG means that the DODAG root can satisfy the connectivity of all hosts and floated otherwise). *MOP* field defines the used mode of operation that is set by the DODAG root and defines. There are 4 modes 1) No Downward routes maintained by RPL, 2) No storing mode of operation, 3) Storing Mode of Operation with no multicast support, and 3) Storing Mode of Operation with multicast support). The *Prf* field defines how preferable the root node is compared to other root nodes. DTSN field is the destination advertisement trigger sequence number field that is a number maintained by the node issuing the DIO message and guarantees

<b>RPLInstanceID</b>				<b>Version Number</b>	<b>Rank</b>	
<b>G</b>	<b>0</b>	<b>MOP</b>	<b>Prf</b>	<b>DTSN</b>	<b>Flags</b>	<b>Reserved</b>
<b>DODAGID (128 bit)</b>						
<b>Option(s)</b>						

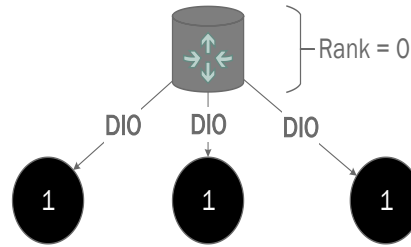
**Figure 2.6: DIO Message**

the freshness of the message. The *DODAGID* field is used to identify node, and the *flags* and *reserved* fields are generally initialized to zero by the senders and are ignored by the receivers. The *option* field is illustrated in Figure 2.7 in which the first two bytes represent the option type, the *DIOIntMin*, *DIOIntDoubl* and *DIORedun* fields are used for the *Trickle timer*, *MaxRankIncrease* defines an upper limit for the *Rank*, *MinHopIncrease* stores the minimum increase of the rank between a node and any of its parent nodes, OCP is the objective code point that identifies the OF and the metrics that the nodes will use to choose their parents, and the last two fields define the *life time* that is used as default for all RPL routes.

Type	Opt Length	FLAGS	A	PCS	DIOIntDoubl.
DIOIntMin	DIORedun	MaxRankIncrease			
MinHopIncrease		OCP			
Reserved	Def Lifetime	Lifetime Unit			

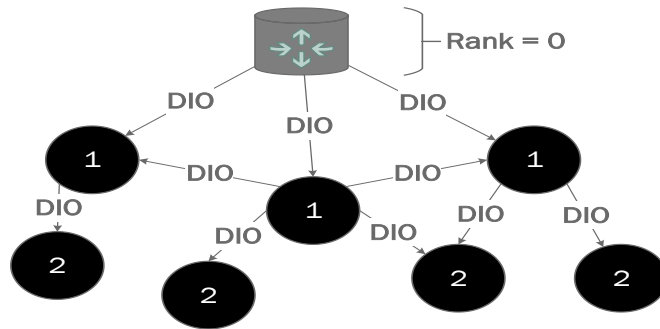
**Figure 2.7: RPL Control Message Option Field**

The construction process starts at the border router when it broadcasts DIO messages to its neighbors. An appropriate parent selection is done based on the metrics and the constraints defined by the objective function. At this time, a route gets established between the current nodes and the DODAG root that will be their preferred parent as shown in Figure 2.8 which assumes the hop-count metric is being used.



**Figure 2.8: Root broadcasts DIO messages to its one hop neighbors. They calculate its ranks according to their distances to the root**

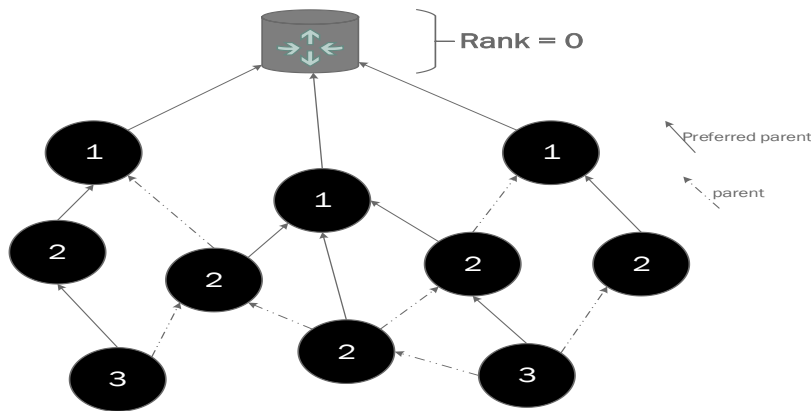
Upon receiving DIO message, each node calculates its own rank and updates the DIO message by adding its rank then broadcasts DIO to its neighbors as shown in Figure 2.9. Indeed, the 3 nodes below the root broadcast DIO messages to their neighbors, which in turn calculate their ranks and select their preferred parents. Nodes that have rank equals to 1 and receive message with rank 1 will not recalculate its rank since they are already in the minimum distance to the root.



**Figure 2.9: node that receives DIO message will calculate its rank then broadcasts new DIO containing its rank. If a node receives a DIO message with rank less than or equal to its rank, it will ignore this message in order to illuminate the loop.**

The process of broadcasting the DIO messages and selecting the set of its parents and the preferred one continues till it hits the leaf nodes. This process will build the DODAG topology from the root node to the leaf nodes as shown in Figure 2.10. Therefore, the use of DIO messages is mainly to establish the paths for upward routing (multipoint to point). During the

construction process of DODAG, each node will have a routing entry towards its parent or multiple parents (depending on the objective function in a hop-by-hop fashion) and the leaf nodes can send a data packet to root router by forwarding the packet to its immediate preferred parent. RPL supports constraint-based routing where constraints may be applied to both link and nodes. It includes the loop avoidance mechanism during the topology changes by using rank-based data path validation mechanisms. When this process finishes, the upward routing paths will be achieved where each node can reach the root by using its collected information (like the preferred parent).



**Figure 2.10:** this is a DODAG tree that could be formed while constructing the RPL network. every node has a set of parents and a preferred one which is the parent that forwards its data through. This figure shows also that when the DIO messages reach the leaf nodes, the tree is fully constructed, and every node is able now to send its data to the border router. Example, the node with rank 3 can send its data to the root, by forwarding them to its directly connected preferred parent with rank 2 then to the node with rank 1 until they reach the root.

For setting up Downward routes from the root node to other nodes which is optional based on MOP in the control messages, each node that joins the DODAG sends upward a unicast *Destination Advertisement Object* (DAO) control message to establish Downward routes as shown in Figure 2.11. This will help the root to have a full view on the graph. RPL supports two modes of downward traffic: *Storing* and *Non-Storing* modes. In the *Non-Storing* case, the packet

will travel all the way to a DODAG root before traveling down to the destination. However, in the storing case, the packet may be directed down towards the destination by a common ancestor of the source and the destination prior to reaching a DODAG root. After that, if a node has not received any DIO and has not joined any DODAGs, it can request DODAG information by sending DIS messages periodically (Trickle Timer) to its neighbors. So, the received nodes send DIO messages to this node, where it can calculate its own rank and join the DAG as shown in Figure 2.12.

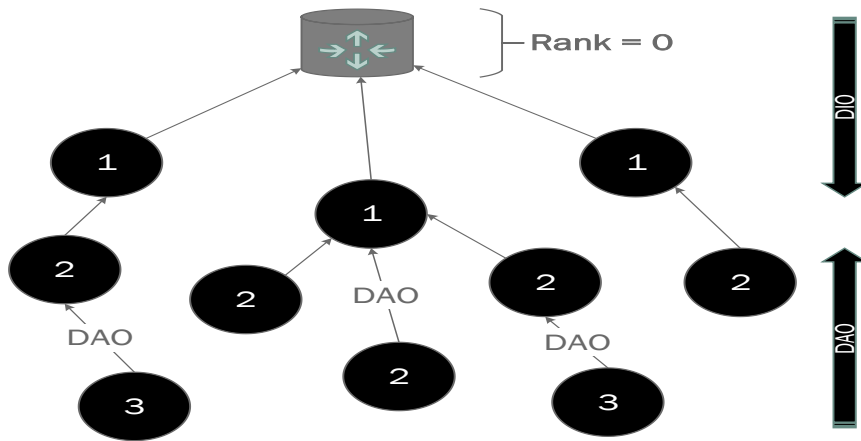


Figure 2.11 This figure shows how the DAO messages are broadcasting bottom up to create the top down routing paths. To the right, there is an up arrow that represents DAO messages exchange, and a down arrow that represents DIO messages exchange.

### 2.2.6. Metrics, Constraints, and Objective Functions:

As we already mentioned, the construction of the DODAG depends on some metrics, constrains, and the objective function. In this subsection, we elaborate further on these features. Because of the type of data traffic, some advanced metrics are required in LLNs, so the routing algorithm can choose the best possible route towards the access point. Moreover, constraints can be used to cut the links or nodes from the DODAG that do not meet certain requirements. Hence,

it is essential to control the alteration rate of the routing metrics in order to avoid path instabilities, which can severely harm LLN performance.

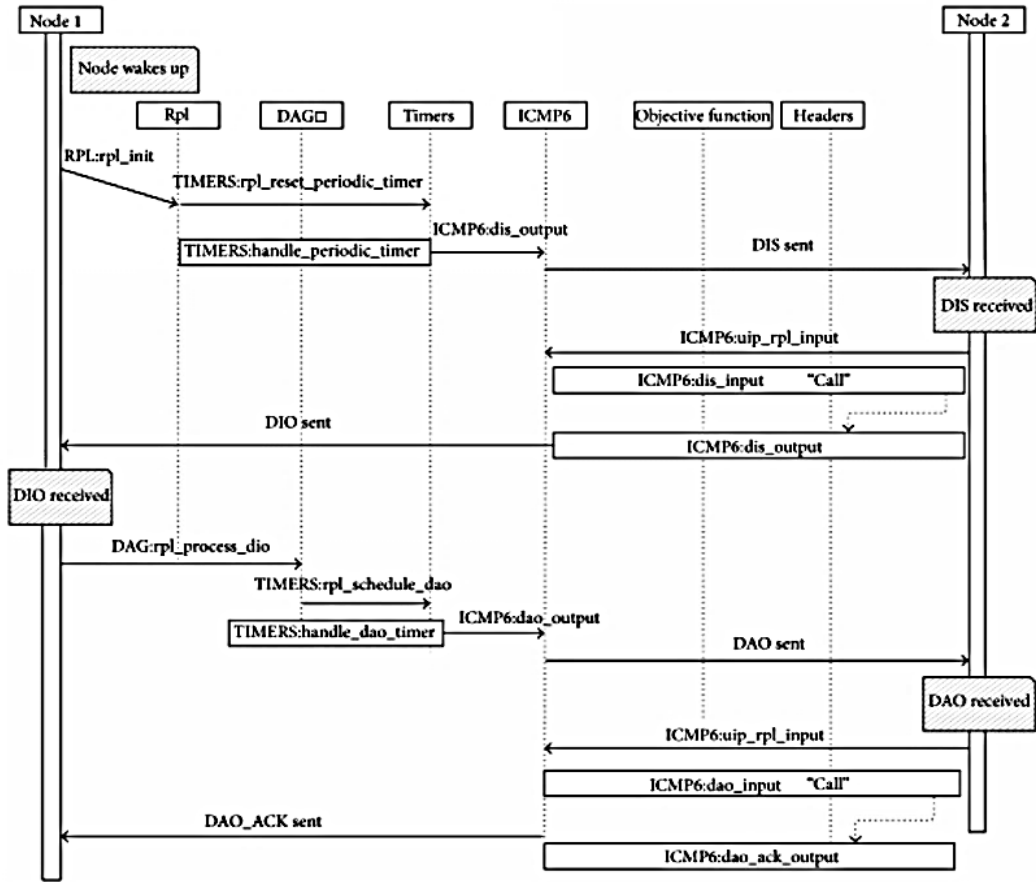


Figure 2.12 Node 1 wants to join the network, it has to send a DIS message to its neighbors (node 2) which replies with DIO messages. Upon receiving the DIO message, node 1 will calculate its rank and reply a DAO message (in case the MOP was not set to 0 i.e. downward route is required) node 2 then acknowledges this DAO message

The objective function is indicated in the DIO message using an objective code point (OCP) which indicates the method that must be used to construct the DODAG. OFs translate the key metrics and constraints into rank, which represents the node distance from a DODAG root, in order to optimize the network topology in a flexible way [10].

### **2.2.7. Loop Detection and Avoidance:**

It is necessary to detect the loops as early as possible to avoid some bad events such as packet dropping or delay in the network. RPL has a simple mechanism to detect and prevent the formation of loop in the network. In this mechanism, Routers multicast DIO messages for topology setup as well as maintenance. Nodes receiving these messages use them for computing their set of parent nodes. In this process, there is always a danger that a node might select its own child as a parent. Therefore, RPL node does not process the DIO messages coming from nodes with higher ranks.

### **2.2.8. Trickle Timer:**

Traditional routing protocols update their routing table periodically. This periodic update mechanism is not useful in RPL as LLNs are resource constraint networks. RPL uses trickle timer mechanism which is adaptive in nature. It controls the sending rate of DIO messages which are responsible for topology formation or its maintenance. The algorithm treats building of graphs as a consistency problem and makes use of trickle timers to decide when to multicast DIO messages. When the network gets stable the interval of the trickle time increases and whenever the inconsistencies are increased the interval decreases. Higher value of trickle timer results into less transmission of DIO control messages and less value produces more DIO control messages [33].

## **2.3. Security in IoT:**

Implementing complex security algorithm is a core attention for LLNs, where it may be economically or physically impossible to include sophisticated security provisions in an RPL protocol. Therefore, the security features are optional to implement in RPL. The implementation



could support integrity and confidentiality. Hence, it should specify which security mechanisms are supported. RPL supports three security modes which are:

1. **Unsecured mode:** The RPL messages are sent without additional security mechanisms. This mode doesn't implicate an unsecured network. it could be using some different security primitives in other network layer to meet the required security.
2. **Preinstalled mode:** in this mode, nodes that want to join an RPL instance should have preinstalled keys that enable them to process and generate secured RPL messages.
3. **Authenticated mode:** It is similar to preinstalled mode, where nodes have preinstalled keys, but the preinstalled keys may only be used to join the RPL Instance as a leaf. Joining as router needs obtaining a key from an authentication authority.

IETF specifies CCM – counter with CBC-MAC (cipher block chaining – message authentication code) as the cryptographic basic for RPL security [35].

To build a secure IoT topology, RPL uses four main ICMPv6 control messages which are secure versions for {DIO, DIS, DAO, and DAO-ACK}.

Types	Code	Checksum
<b>Security</b>		
<b>Base</b>		
<b>Option(s)</b>		

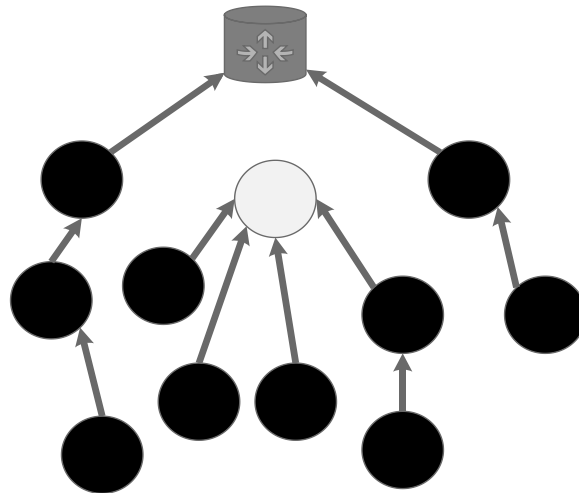
**Figure 2.13: The format of RPL control message with its security field**

Figure 2.13 shows the format of a secure RPL control message which contains a security field [11] [12] indicating the level of security and the cryptographic algorithms employed to process security for the message. By applying the security field to the message, the network will support Integrity 1) integrity which is used to make sure that nobody in between node A and B changed

some parts of the shared information, 2) data authenticity which is used to make sure that node really communicates with the specified node, 3) semantic security and protection against replay attacks, 4) confidentiality which is used to make sure that nobody in between node A and B is able to read what data or information is sent between the two nodes and 5) key management.

In this section, we present some attacks against RPL network, and proposed counter measures, and their consequences on network parameters. RPL does not have the self-healing capacity against most of these attacks.

**1. Selective Forwarding attack:** it occurs when a malicious node removes some packets without forwarding them to the sink node. If this attack destroys or removes all packets, its name will be Black Hole attack. One counter measure against this attack is to create a disjoint path or dynamic path between parent and children. Figure 2.14 illustrates a malicious node that drops all the packets.



**Figure 2.14: Black hole Attack, the white node drops all or some received packets.**

**2. Sinkhole Attack:** in this attack a malicious node attempts to attract the most possible routes to control over most of the data transmitted through the network. The adversary will appear to others as being very attractive by presenting optimal routes. One solution for this attack is

to use the rank authentication technique that relies on one-way hash technique. The root node starts to generate hash value by picking random value and broadcasting it in DIO message. All nodes calculate the hash value using previous received one and again broadcast it using DIO message. Assumed that malicious node doesn't calculate the hash value, it simply broadcasts received DIO message. Each node stores the hash value received by its parent along with number of hops in the path. This attack is shown in Figure 2.15 where most of the packets are forwarded through the white node which is a malicious node.

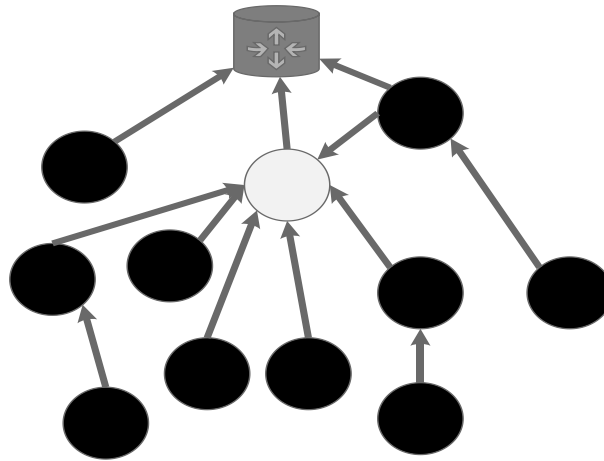
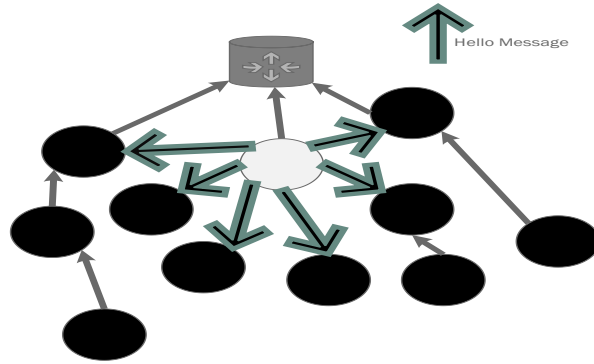


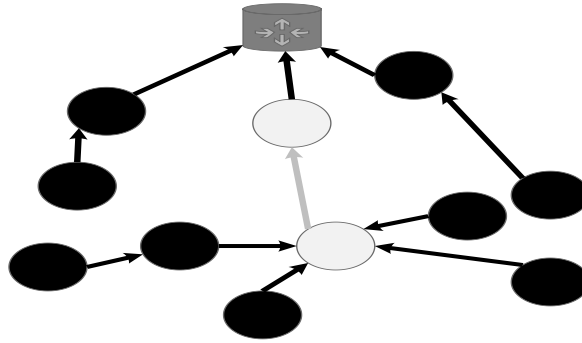
Figure 2.15: Sinkhole Attack, the white node attracts most of the nodes to select it as preferred parent

- Hello Flood Attack:** a malicious node can introduce itself as neighbors to many nodes by broadcasting hello messages with strong metrics to join the DODAG as shown in Figure 2.16.



**Figure 2.16: Hello Flood Attack, the white node broadcasts hello message to many nodes in order to overwhelm the network.**

4. **Wormhole Attack:** It is produced by at least 2 malicious nodes that can communicate with each other through a different frequency than the network. So, they operate in a way that communication between them is still discrete relative to other nodes. One of these nodes is placed near the border router and the other a little further, when one receives packets it transmits them directly to the other without passing through the normal path. In this way, the malicious nodes can manipulate the packets. One solution is by using the Markle tree authentication where in RPL the tree construction starts from root to leaf nodes and Markle tree construction starts from leaf node to root [28]. It uses ID of node and public key for calculation of hash. Each parent is identified by its children. Authentication of any node begins with the root node up to the node itself. If any node failed to authenticate, then children nodes avoids the wrong parent selection.



**Figure 2.17: Wormhole Attack, the communication between the 2 white nodes are happened using different frequencies**

5. **Rank Attack:** in this attack a malicious node can use a fake rank in order to attract the others and make children as much as it can where most of the traffic will pass through it. This will lead to an unoptimized paths, may form loops, Increase the delay in the network, and more control overheads. One solution to prevent this attack is proposed by version number and authentication (VeRA) where a security service was achieved for preventing the misbehaving node from decreasing Rank values for attack purpose. VeRa prevents publishing an illegitimate decreased Rank by generating the hash chaining using random number chosen by root node [29].
6. **Good/Bad Mouting Attacks:** In this attack, a malicious node broadcasts fake information about other node N for the sake of letting nodes to forward packets through N (good mouting), or in order to illuminating this node N (bad mouting).
7. **Version Number Attack:** In RPL, the version number is carried in the control messages and is used as a global repair operation indicator when the DODAG route changes it. when this number is modified by the DODAG route, all the nodes start to exchange control message to build a new topology. RPL doesn't specify a mechanism to protect the version number from

illegal modification. Therefore, a malicious node can still modify this number and force other nodes to start exchanging control messages and overwhelm the network [13].

**8. Whitewashing attacks:** A malicious node can disappear and rejoin the application to wash away its bad reputation.

**9. Ballot-stuffing attacks:** a malicious node can boost the reputation of other node by providing good recommendation for it in order to increase the chance of this bad node being selected as a service provider.

**10. Sybil or identity attacks:** an attacker copies the identities of a valid node onto another physical node. This can be used in order to gain access to a larger part of the network in order to overcome voting schemes. In this attack, an attacker uses several logical entities on the same physical node.

Also, there are some other attacks that can still be launched in RPL protocol. Next chapter surveys some related works to mitigate these attacks using kind of trust relationships between nodes.

## CHAPTER 3

### APPROACHES TO MITIGATE SOME ATTACKS IN RPL

Some trust management approaches were proposed to mitigate the attacks that can be launched in RPL network. In this chapter, we survey some of these approaches by summarizing and analyzing them [14], [15], [16], [17], [18], [19], [20].

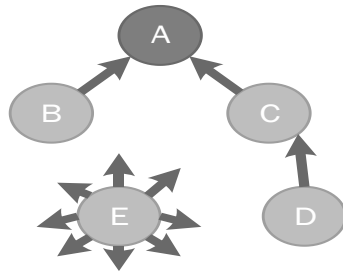
#### **3.1. Towards a trust computing architecture for RPL in CPS [14]**

Cyber-Physical Systems (CPS) are integrations of computation, networking, and physical processes that use RPL as a routing protocol. Since the RPL standard doesn't propose an efficient way to ensure the integrity and confidentiality of message, it is necessary to develop an appropriate security management system for LLNs using RPL. Furthermore, because applying a cryptographic algorithm to the messages will occupy most size of the memory and take many CPU cycles (since the devices are constrained), [14] proposes to use trusted platform module (TPM) as a coprocessor installed in each device, in order to minimize the resource usage and provide most of the security features. The TPM is a hardware component that securely stores digital keys, certificates, and passwords. It is designed in a way (its design is beyond the scope of our work) to protect key operations and other security tasks that would affect negatively the performance of the network.

To establish a trusty network, they suggested to develop resource constrained devices that include TPM chips, where each TPM must have up to 5 symmetric-keys stored in the write only section (Identified-Establishment Keys, IKs). Also, The TPM contains asymmetric key-pair used to securely pass the symmetric group key (GK) that is used by the RPL network and stored in the TPM. If a new node  $N$  wants to join the network (DODAG tree), it has to obtain the appropriate

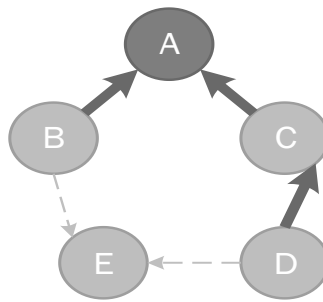
GK, so it will be able to decrypt the contents of the RPL control messages. Therefore, it must follow the below procedures:

- First, node *N* broadcasts a *GKRequest* message containing its public key (*PubKey*) and a random number (*RanNum*) generated by the TPM as shown in Figure 3.1



**Figure 3.1: E broadcasts GKRequest to all its neighbors in order to get the group key**

- Nodes that receive this message encrypt *RanNum* leading to *EncRanNum* using randomly chosen IK, then respond as shown in Figure 3.2 with a *GKChallenge* message encrypted by *PubKey* and containing *EncRanNum* and another random number generated locally.



**Figure 3.2: B and D reply by sending a GKChallenge messages to E**

- Upon receiving any response, *N* decrypts the message using its private key. Then, it proceeds to decrypt *EncRanNum* with the IKs stored in its TPM until getting its *RanNum*. If none of the IKs leads to its *RanNum*, the node sending this response is deemed untrustworthy and further communication with it is ignored. On the other hand, if a valid decryption is obtained (getting *RanNum*), the node wishing to join now trusts that node to



provide a valid GK. At this time, N will send a *GKChallenge* to that node and same process will happen. A valid decryption at that moment proves trustworthiness of node N.

- Using this procedure, mutual trust of both nodes is established, and the new node gets the GK, so it can decrypt all the secure control messages.

Since an attacker may decipher the GK using differential cryptanalysis, the approach proposed a way to change the GK periodically or on demand. That can be achieved by generating a group key update message from the root towards the other nodes.

The main limitation in [14] can be summarized as following. The approach proposed to use an extra hardware chip (TPM) where, a TPM must be installed in each device. In addition, the TPM stores only 5 keys (because of its limited storage), so an adversary can easily decipher these keys by differential cryptanalysis (a mechanism to update these keys periodically like the GK is needed). Also, this proposed solution didn't take into consideration some attacks like black whole attack when a node drops the packets instead of forwarding them, denial of service attack and other attacks that might exhaust the network. Moreover, this approach didn't make any modification to the RPL protocol, but it just added a preconstruction step, i.e. before starting the construction of the DODAG. So, this work can be a good step to other work in order to have a better encryption/decryption process. Adding to that, the proposed approach was not evaluated using some RPL simulator, so there is no evaluation available regarding energy consumption or delay in the network.

### **3.2. Trust-based service management for social IoT systems [15]**

A social internet of things (IoT) system can be viewed as a mix of traditional peer-to-peer (P2P) networks and social networks, where each node (thing) will have a level of trustworthiness for any other node even they are not neighbors. Some misbehaving devices may perform

discrimination attacks for their own gain at the expense of other IoT devices which provide similar services. To address this problem, an adaptive trust management protocol for social IoT was proposed in [15]. In social IoT systems, devices are owned by users who have social relationships between each other. These social relationships are translated into social relationships between IoT devices. The proposed adaptive trust management is interaction-based as well as activity-based, meaning that the trust value is updated dynamically upon an interaction event or activity. Two nodes involved in a direct interaction activity can directly observe each other and update their trust assessment. They also exchange their trust evaluation results toward other nodes as recommendations.

From the many available social trust metrics, the proposed approach chose *honesty*, *cooperativeness*, and *community-interest* as the most striking metrics for characterizing social IoT systems. The *honesty* trust property represents whether a node is honest or not, where a node relies on direct evidence and indirect evidence to evaluate the honesty trust property of another node. The *cooperativeness* trust property represents whether or not the trustee node is socially cooperative with the trustor node, where a node can evaluate the cooperativeness property of other nodes based on social ties and select socially cooperative nodes in order to achieve high application performance. The *community-interest* trust represents whether or not the trustor and the trustee nodes are in the same social groups.

The adaptive trust management is a continuing process which iteratively aggregates past information and new information. The new information includes both direct observations and indirect recommendations. The trust assessment of node  $i$  towards node  $j$  at time  $t$  is denoted by  $T_{ij}^X(t)$  where  $X = \textit{honesty}, \textit{cooperativeness}, \textit{or community-interest}$ .  $T_{ij}^X(t)$  is updated when node  $i$  interacts with node  $j$ , as follows

$$T_{ij}^X(t) = (1 - \alpha)T_{ij}^X(t - \Delta t) + \alpha D_{ij}^X(t) \quad (3.1)$$

Where  $\Delta t$  is the time interval between two consecutive interaction,  $D_{ij}^X(t)$  is the direct observation on  $i$  towards  $j$ , and  $\alpha$  is to weigh the direct observation.

$D_{ij}^X(t)$  for each value of  $X$  is calculated as below:

- $D_{ij}^{honesty}(t)$ - Direct *honesty* trust refers to the belief of node  $i$  that node  $j$  is honest based on node  $i$ 's direct interaction experiences toward node  $j$  at time  $t$ . First, node  $i$  detects bad-mouthing/ballot-stuffing attacks by node  $j$  by comparing node  $j$ 's recommendation toward another node, say node  $q$ , with the trust value of node  $i$  toward node  $q$  itself. Node  $j$ 's recommendation toward  $q$  is just  $D_{jq}^{honesty}(t)$  if node  $j$  is an honest node not performing attacks; otherwise, it can be 0 (or 1) if node  $j$  is a dishonest node performing bad-mouthing attacks against (or ballot-staffing attacks for) node  $q$ . Node  $i$ 's trust toward node  $q$  on the other hand is just  $D_{iq}^{honesty}(t)$  kept by node  $i$ . If the percentage difference relative to  $D_{iq}^{honesty}(t)$  is higher than a threshold, it is considered suspicious and thus a negative honesty experience.
- $D_{ij}^{cooperativeness}(t)$  Each device keeps a list of its owner's friends which may be updated dynamically by its owner.  $D_{ij}^{cooperativeness}(t)$  is computed as the ratio of common friends between  $i$  and  $j$ , i.e.  $\frac{(friends(i) \cap friends(j))}{(friends(i) \cup friends(j))}$ , where  $friends(i)$  denotes the set of friends to the owner of node  $i$ .
- $D_{ij}^{community-interest}(t)$  This trust property provides the degree of the common interest or similar capability of node  $j$  as evaluated by node  $i$  based on direct observations at time  $t$  upon encountering. Each device keeps a list of its owner's communities/groups of interest

which may be changed dynamically.  $D_{ij}^{community-interest}(t)$  is computed as the ratio of common communities of interests between nodes  $i$  and  $j$ , i.e.,  $\frac{(communities(i) \cap communities(j))}{(communities(i) \cup communities(j))}$ , where  $communities(i)$  denotes the set of communities of interests to the owner of node  $i$ . When node  $i$  and node  $j$  directly interact with each other, with permission granted from their owners they also exchange their service and device profiles, so node  $i$  can validate whether node  $j$  and itself are in a community/group. Therefore, the direct observation of *community-interest* trust will be close to actual status.

Moreover, when the node  $i$  interacts with node  $k$  ( $i \neq j$ ) that has a prior interaction experience with node  $j$ .  $k$  will serve as a recommender to provide its trust recommendation about node  $j$  to node  $i$ . so,  $i$  will update its trust value toward  $j$  as follows:

$$T_{ij}^X(t) = (1 - \gamma)T_{ij}^X(t - \Delta t) + \gamma R_{kj}^X(t) \quad (3.2)$$

where  $R_{kj}^X(t)$  is the trust value that comes from  $k$  as a recommendation for  $j$ ,  $T_{ij}^X(t - \Delta t)$  is the old trust value for  $i$  toward  $j$ , and  $\gamma$  is used to weigh the new recommendation versus past experience and to consider trust decay over time.

At the end each node will have  $T_{ij}^{honesty}(t)$ ,  $T_{ij}^{cooperativeness}(t)$ , and  $T_{ij}^{community-interest}(t)$ , so it has to comprise them in a way to get  $T_{ij}(t)$  that is the final trust value of  $i$  toward  $j$ . But  $T_{ij}(t)$  is a trust formation issue and depends on the trust requirement of the IoT application.

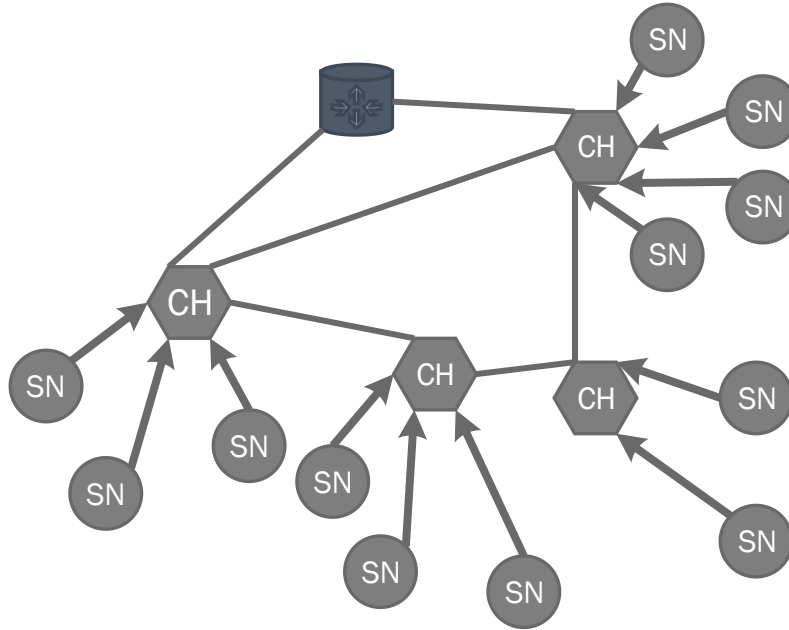
This approach [15] took into account the following malicious attacks, self-promoting attacks, whitewashing attacks, discriminatory attacks, bad-mouthing attacks, and ballot-stuffing attacks. However, a malicious node can still perform Sybil and identity attacks, and in general can perform communication protocol attacks to disrupt IoT network operations. Also, an attacker can perform other bad behavior such as selective attacks to damage the network. In addition,

each node has to store the 3 trust values of all other nodes, so if we have a large IoT network, we cannot apply this trust management system since the storage requirement is still excessive for IoT devices (limited storage). Also, this approach was not evaluated on RPL based network and was not incorporated in RPL control messages to construct the DODAG tree. Adding to that, this proposal was made for social IoT, so it used only the social relationships between nodes, without any addition metrics like energy aware, or hop counts... or even knowing if the node is working well or not (i.e. whether the node is forwarding or dropping the packets).

### **3.3. Hierarchical trust management for wireless sensor networks and its application to trust-based routing [16]**

A cluster-based hierarchical trust management protocol for wireless sensor networks was proposed in [16] to effectively deal with selfish or malicious nodes. It considers multidimensional trust attributes derived from communication and social networks to evaluate the overall trust of a sensor for wireless sensor network (WSN) applications wherein both social trust and quality of service trust are important for mission execution. This trust management proposal is made for cluster-based WSN consisting of multiple clusters, each with a cluster head (CH) and a number of sensor nodes (SNs) as shown in Figure 3.3. SN forwards its sensor reading to its CH through SNs in the same cluster. The CH in its turn forwards the data to the base station. Because of this hierarchy structure in the WSN, the trust management protocol is conducted using periodic peer-to-peer trust evaluation between two SNs and between two CHs. At the SN level, each SN is responsible to report its peer-to-peer trust evaluation results towards other SNs in the same cluster to its CH which applies statistical analysis and performs CH-to-SN trust evaluation towards all SNs in its cluster. Trust metrics were formed based on both social trust and QoS trust to consider the effect of both aspects of trust on trustworthiness. --Social trust

may include friendship, honesty, privacy, similarity, betweenness centrality, and social ties (strengths), where QoS trust may include competence, cooperation, reliability, task completion capability--.



**Figure 3.3: Cluster Based Wireless Sensor Network consisting of multiple clusters.**

In this approach, the trust management system maintains two levels of trust: sensor node and cluster head. Where each SN evaluates other nodes in the same cluster, and each cluster head evaluates the SNs in its cluster and other CHs. These evaluations will be accomplished periodically based on direct or indirect observations via snooping or overhearing. the procedure will follow the steps below:

- Each SN evaluates other nodes in the same cluster and shares its trust evaluation results with other nodes in the same cluster and its CH.
- Each CH performs trust evaluation toward all SNs within its cluster and shares this trust evaluation results with other nodes in the WSN and with a “CH commander” which may

reside on the base station, or on an elected CH. The CH commander performs trust evaluation toward all CHs in the system. Therefore, it is a centralized trust management.

The SNs evaluate each other using four different trust components which are: intimacy, honesty, energy and unselfishness. The trust value is evaluated as follows:

$$T_{ij}(t) = w_1 T_{ij}^{intimacy}(t) + w_2 T_{ij}^{honesty}(t) + w_3 T_{ij}^{energy}(t) + w_4 T_{ij}^{unselfishness}(t) \quad (3.3)$$

Where  $w_1 + w_2 + w_3 + w_4 = 1$  and  $T_{ij}^X(t)$  is calculated by relying on direct and past observations when  $i$  and  $j$  are neighbors and on recommendation when  $i$  and  $j$  are not neighbors.

$$T_{ij}^X(t) = \begin{cases} (1-a)T_{ij}^X(t-\Delta t) + aT_{ij}^{X,direct}(t) & \text{if } i \text{ and } j \text{ are neighbors} \\ \text{avg}_{k \in N_i} \{ \gamma T_{ij}^X(t-\Delta t) + (1-\gamma)T_{kj}^{X,recom}(t) \} & \text{Otherwise} \end{cases} \quad (3.4)$$

The components of equation (3.4) are described as following:

- $T_{ij}^{intimacy,direct}(t)$ : Measures the level of interaction experiences. It is computed by the number of interactions between nodes  $i$  and  $j$  over the maximum number of interactions between node  $i$  and any neighbor node over the time period  $[0, t]$ .
- $T_{ij}^{honesty,direct}(t)$ : Refers to the belief of node  $i$  that node  $j$  is not compromised based on node  $i$ 's direct observations toward node  $j$ . It can be a binary quantity, 0 or 1,
- $T_{ij}^{energy,direct}(t)$ : This indicates the percentage of node  $j$ 's remaining energy that node  $i$  directly observes at time  $t$ .
- $T_{ij}^{unselfishness,direct}(t)$ : This provides the degree of unselfishness of node  $j$  as evaluated by node  $i$  based on direct observations.

The above values can be calculated when nodes  $i$  and  $j$  are neighbors. If they are not neighbors, node  $i$  will use the recommendation given by other nodes.

The approach in [16] is made for only cluster based WSN consisting of multiple clusters, so, it can't be applied to any WSN. Also, this trust management is centralized where all the evaluations made by the nodes will be sent to the base station or to the CH commander leading to more energy consumption especially in the CHs. Adding to that, the different metrics used in their approach are calculated using the energy as a parameter. As a consequence, if a normal node is surrounded by selfish nodes, it will consume more energy, and it can be considered as untrusted while it is trusted. Moreover, this approach didn't take into accounts if a malicious node was dropping or forwarding the packets i.e. it doesn't avoid black/grey hole attacks.

### 3.4. Trust-based RPL for the Internet of Things [17]

The approach proposed in [17] believes that using TPM (trusted Platform Model) to ensure trustworthiness between nodes is not sufficient. Indeed, a malicious node could participate in constructing the RPL topology. Therefore, to overcome this issue, it proposes to improve the RPL protocol by adding a new trustworthiness metric during RPL construction and maintenance. This metric represents the level of trust for each node in the network that is calculated using *selfishness, energy and honesty* components (i.e. each node calculates a set of trust values of its direct neighbors). It allows a node to decide whether or not to trust the other nodes during the construction of the DODAG. To meet this purpose, a new node metric was added in the RPL control messages then a new objective function that uses this metric was introduced to select parent and to calculate Rank (*off-load computations into a Trusted Platform Module (TPM)*). The steps below describe the procedure of the RPL construction algorithm:

1. The border router declares itself as a floating root and hence has no parent.
2. Each node  $i$  evaluates the trust value of its one hop neighbor node  $j$  at time  $t$ .

$$T_{ij}(t) = w_1 T_{ij}^{honesty}(t) + w_2 T_{ij}^{energy}(t) + w_3 T_{ij}^{unselfishness}(t) \quad (3.5)$$



where  $w_1 + w_2 + w_3 = 1$  and  $T_{ij}^X(t)$  is calculated according to the below equation where  $x = \{honesty, energy, unselfishness\}$ .

$$T_{ij}^X(t) = (1 - \alpha)T_{ij}^X(t - \Delta t) + \alpha T_{kj}^{X,direct}(t) \quad (3.6)$$

3. When node  $i$  receives DIO messages from its neighbors, it computes the new trust value for each neighbor node  $j$ . This new trust value is the average of its trust value calculated according to the above equations and all trust values received for that neighbor  $j$ . The obtained result represents the final trust value for neighbor  $j$ , and it will be used to select the set of parents and the preferred parent.

$$T_{j\ Final} = \frac{\sum^m T_{ki}}{m} \quad (3.7)$$

Where  $m$  represents the number of nodes from which node  $i$  received trust values.

4. Node  $i$  calculates its new trust value that is equal to the average of all received trust values for itself.
5. Each node  $i$  computes the path cost for each reachable neighbor  $j$ . This cost represents the cost of node  $i$  to reach border router through node  $j$ .
6. After computing the trust values, node  $i$  is now able to select its set of parents and choose the appropriate one based on  $T_{j\ Final}$  values.
7. Finally, node  $i$  calculates its rank. The calculation starts at the border router, where it sets its rank to  $MinHopRankIncrease$ . Then each node  $N$  calculates its Rank  $R(N)$  as the sum of the Rank of its preferred parent and  $rank\_increase$ .

$$\begin{cases} R(N) = R(P) + rank_{increase} \\ rank_{increase} = step + MinHopRankIncrease \\ step = T_{j\ Final} * 100 \end{cases} \quad (3.8)$$

8. When a node has computed trust values, has selected the preferred parent and calculated the Rank, it updates its metric container (trust values of its neighbors and itself), and sends its own DIOs to its neighbors.

The selected path that a node chooses in [17] could be the longest, because the hop count and ETX were not considered as trust metric. Therefore, the network lifetime will decrease where more energy will be consumed during the transmission of packages. Also, the nodes' behaviors were not analyzed, where a malicious node may drop some of the packets and forms a blackhole attack. Moreover, in this approach, each node needs to store trust values for all its neighbors. Hence if a node is surrounded by many nodes, it will store the trust values of all these neighbors and that will lead to a storage problem.

### 3.5. Design of primary and composite routing metrics for RPL [18]

Formulas for evaluating the RPL routing metrics were proposed in [18], they aim to enrich the routing metric set by a new routing metric which targets the detection and avoidance of malicious/malfunctioning nodes.

In wireless sensor network (WSN), losses may occur such as the black/grey hole attack, during which a node refuses forwarding all/part of the traffic acting either selfishly or maliciously, even though it acknowledges the reception of the traffic. In order to defend against these attacks, a trust-related metric was defined as follows. Each node, after transmitting a packet to a neighbor, enters the illegal mode to listen whether the selected parent has actually forwarded its packet, thus building trust knowledge. So, the packet forwarding indication (PFI) metric was defined such that a node will choose to set its preferred parent. Assume that  $sf$  packets were actually forwarded and  $ff$  packets failed to be forwarded, then the path weight  $w(a) = \log\left(\frac{1}{P_{succ(a)}}\right) =$

$\log\left(\frac{sf_a+ff_a}{sf_a}\right)$  for any link  $a$ . Therefore, a node will choose its preferred parent by selecting the paths that have the *minimum weights* ( $w(a)$ ). Since this metric alone is not adequate, this approach proposes to combine it with the *hop count* metric or the *energy metric*. The combination of these metrics can be computed additively by adding both weighted metrics or lexically.

The weakness of the approach in [18] is the fact that each node takes a decision based only on its own knowledge. However, if this node misbehaves, it will choose a failing path rather than a good trusted path. Also, this paper only considers the black/grey hole attacks.

### 3.6. Using trust management to defend against routing disruption attacks [19]

this approach addresses the problem of routing disruption attacks in cognitive radio network (CRN). In this attack, a malicious node intentionally drops data packets to consume more network resources. Therefore, it proposes a trust a management model to mitigate such attacks. The main idea is to use a quantifiable value (trust) to evaluate the mutual relationship between two nodes. In CRN routing context, the concept of trust was defined as a representation of the degree that the nodes honestly forward data packets to the second hop. Whereby, a node uses the statistics of the next hop forwarding behaviors to calculate the trust value of that neighbor. This trust value is denoted by  $T_{ij}(t)$  that is an estimate between 0 and 1 where 0 is the lowest trust value. The calculation of this value is shown below:

When  $i$  is observing  $j$ , the number of successfully forwarding observed is denoted by  $r_{ij}(t)$  and the number of failed forwarding observed is denoted by  $s_{ij}(t)$ . So, the probability that neighboring  $j$  forwards data packets to the next hop is:

$$p = \frac{r_{ij}(t)+1}{s_{ij}(t)+r_{ij}(t)+2} \quad (3.9)$$

This probability will be equal to link quality, then the evaluation function  $\tau(t)$  that is equal to  $\tau(t) = 1 - \frac{|\bar{p}-p|}{p}$  was evaluated. The value  $T_{ij}(t)$  reflects the trust history of neighboring  $j$  before time  $t$ , and the evaluation function  $\tau(t)$  reflects the recent forwarding behavior of neighboring  $j$  during the observed time step.

$$T_{ij}(t + 1) = (1 - w)T_{ij}(t) + w\tau(t) \quad (3.10)$$

When the data packet of node  $i$  needs to be sent to node  $j$ , a route should be decided between the source and destination pair. The route selection could be performed in proactive manner or on-demand manner.

The main limitations of [19] is that it only considers the disruption attacks like the black/grey attack. Also it was proposed for cognitive radio network and not for IoT. It just uses statistics evaluation by checking if the next hop forwards or drops the data packets, so a malicious node still launch Sybil or clone attacks. Also, a node will take a decision based only on its own knowledge without checking how other nodes evaluate a specific node it wants to forward data through it.

### **3.7. Link Reliable and Trust Aware RPL (LT-RPL) for IoT**

The approach proposed in [20] aims to ensure trust among nodes and to provide quality of service during the construction and maintenance of the network routing topology. So, it proposes LT-RPL protocol that is a link reliable and trust aware model for RPL protocol, where this protocol follows a multidimensional approach to enable an accurate trust value computation for IoT entities. To build their trust management system, this approach considers an IoT network consisting of several groups made up by a set of smart objects randomly deployed in a circular network. Each IoT device has trust manager which is an entity in charge of assessing the trustworthiness value of another object or link. In this approach, the design of LT-RPL consists

of several phases which are *information gathering, trust composition, trust database and trust application*.

In the information gathering phase, the trust related information is collected regarding the nodes' behavior as well as link's indicators (using direct observation and recommendation).

In the trust composition phase, the trust manager computes the trust level for each node specifically the node related trust (NT) and the link related (LT) as follow:

$$T(e_i \rightarrow e_j)_t = w_1 NT(e_i \rightarrow e_j)_t + w_2 LT(e_i \rightarrow e_j)_t \quad (3.11)$$

Where  $T(e_i \rightarrow e_j)_t$  denotes the trusted value calculated by node  $i$  towards node  $j$  at time  $t$ ,  $NT(e_i \rightarrow e_j)_t$  is the node related trust that can be calculated based on direct observation of its one hop neighbors' behavior and indirect observation of other nodes as

$$NT(e_i \rightarrow e_j)_t = w_d NT(e_i \rightarrow e_j)_t^d + w_r NT(e_i \rightarrow e_j)_t^r \quad (3.12)$$

The direct trust is calculated by considering both node cooperativeness and node competence at time  $t$  as

$$NT(e_i \rightarrow e_j)_t^d = NT(e_i \rightarrow e_j)_t^{coop} * NT(e_i \rightarrow e_j)_t^{comp} \quad (3.13)$$

When it comes to the link trust  $LT(e_i \rightarrow e_j)_t$ , this value will be calculated as

$$LT(e_i \rightarrow e_j)_t = LT(e_i \rightarrow e_j)_t^{qual} * LT(e_i \rightarrow e_j)_t^{perf} \quad (3.14)$$

where  $LT(e_i \rightarrow e_j)_t^{qual}$  imitates the belief that the connecting link is good enough to respect the QoS guarantees. It is measured by ETX and packet reception ratio as indicators of the link quality between the entity and its neighbor. While  $LT(e_i \rightarrow e_j)_t^{perf}$  describes the performance of the link based on the packet error ratio and the transmission delay.

After the completion of trust values computation, these values will be stored in the Trust storage phase. Finally, each node sends to its neighbors the value of its rank. Once received the evaluating node  $e_{join}$  checks its record table for the most recent trust values of its  $p \geq 1$  candidate parents  $e_{cand_1} \dots e_{cand_p}$ , already sent by the trust manager. At this time,  $e_{join}$  can calculate its rank according to the trust based OF.  $R(e_{join} \rightarrow e_{cand_q}) = R(e_{cand_q}) + T(e_{join} \rightarrow e_{cand_q})_t$  where  $R(e_{cand_q})$  is the rank value of the candidate parent.

This approach [20] considers only the black/grey hole attack, so any other attack like Sybil of bad/good mouthing attacks cannot be avoided. Also, it assumes that every node should have a trust manager which is an entity in charge of assessing the trustworthiness degree of another object or link. And, each node has to calculate and know the trust values of all other nodes.

The Table below briefly illustrates all the above papers:

<b>Paper</b>	<b>Weaknesses</b>	<b>Strengths</b>
<b><i>Towards a Trust Computing Architecture for RPL in Cyber Physical System [14]</i></b>	<ul style="list-style-type: none"> <li>- extra hardware chip (TPM)</li> <li>- Black or grey hole attack can be launched.</li> <li>- Selective, DoS attack also can happen.</li> <li>- Trust forever</li> </ul>	<ul style="list-style-type: none"> <li>- TPM is useful to store and generate group key.</li> <li>- TPM is very cheap and installing it is easy.</li> <li>- Having a better encryption/decryption process.</li> </ul>
<b><i>Trust-Based Service Management for Social Internet of Things Systems [15]</i></b>	<ul style="list-style-type: none"> <li>- A disruption attack like DoS can still be launched</li> <li>- Each node must store the trust value of every other node.</li> <li>- This work uses only social relationship between nodes.</li> </ul>	<ul style="list-style-type: none"> <li>- self-promoting, whitewashing, discriminatory, bad-mouthing, and ballot-stuffing attacks can be avoided.</li> <li>- many social trust metrics can be used to mitigate some other attacks.</li> </ul>
<b><i>Hierarchical Trust Management for Wireless Sensor</i></b>	<ul style="list-style-type: none"> <li>- it is made only for cluster-based WSN consisting of multiple clusters.</li> </ul>	<ul style="list-style-type: none"> <li>- Each node needs to evaluate only the trust values of its neighbors that are in the</li> </ul>

<p><b><i>Networks and Its Application to Trust-Based Routing [16]</i></b></p>	<ul style="list-style-type: none"> <li>- Centralized management system (energy consumption).</li> <li>- Selective and black hole attacks can be launched in this network.</li> <li>- CH consumes more energy</li> </ul>	<ul style="list-style-type: none"> <li>- same cluster.</li> <li>- The trust value is calculated based on social relationships and energy consumption.</li> <li>- The trust values are not stored in the nodes, but in the CH commander.</li> </ul>
<p><b><i>Trust Based RPL for the internet of things [17]</i></b></p>	<ul style="list-style-type: none"> <li>- Selected path could be the longest. Because in their work, they didn't take the HC metric into account</li> <li>- Also, many attacks like black/selective adversary can attack the network.</li> <li>- This work was not evaluated regarding the energy consumption.</li> </ul>	<ul style="list-style-type: none"> <li>- This work is applicable to RPL protocol.</li> <li>- They used some social relations and energy metrics.</li> <li>- Each node uses its direct observation and recommendations to evaluate the trust value of another node.</li> </ul>
<p><b><i>Design of primary and composite routing metrics for RPL-compliant Wireless Sensor Networks [18]</i></b></p>	<ul style="list-style-type: none"> <li>- Each node takes decision based only on its own knowledge.</li> <li>- Bad/good mouthing, Sybil, and clone attack still can be launched.</li> </ul>	<ul style="list-style-type: none"> <li>- Black or grey hole attack can be avoided.</li> <li>- Node just uses its direct observation towards another node to evaluate the trust value of that node.</li> </ul>
<p><b><i>Using Trust Management to Defend against Routing Disruption Attacks for Cognitive Radio Networks [19]</i></b></p>	<ul style="list-style-type: none"> <li>- They only consider the disruption attacks like the black/grey attack.</li> <li>- Made for cognitive radio network and not for IoT.</li> <li>- They just used statistics evaluation by checking if the next hop forwards or drops the data packets.</li> <li>- a node will take a decision based only on its own knowledge.</li> </ul>	<ul style="list-style-type: none"> <li>- Good approach against disruption attacks.</li> <li>- A node evaluates other node based on its experience.</li> </ul>
<p><b><i>Link Reliable and Trust Aware RPL Routing Protocol for Internet of Things [20]</i></b></p>	<ul style="list-style-type: none"> <li>- Black/grey hole attacks are the only adversaries that were taken into accounts.</li> <li>- every node should have a trust manager</li> <li>- each node needs to calculate the trust values of all other nodes.</li> </ul>	<ul style="list-style-type: none"> <li>- They use both node related and link related trusts to calculate the rank value of each node.</li> <li>- The computation of the trust values is bounded by the trust manager of the node.</li> </ul>

### 3.8. Summary

The proposed approach in [14] uses the TPM in order to overpass the misbehaved node. It assumes that a node is malicious if it is not authorized (i.e. it does not have a group key to decrypt the control messages). This approach cannot be an efficient solution for many attacks, where it can only detect the nodes that are not authorized. But it can be useful as a first step when building a new trust management system. This step helps to have a better and less energy consumption encryption/decryption processes.

The main limitation of the trust-based service management for social IoT that was proposed in [15] is that every node must store 3 trust values for all other nodes, so this approach cannot be applied for a big IoT system. Hence, to improve this approach a node can only store the evaluation of its neighbors by making social relationships with them. In this case, the node can know the trust evaluation of nodes that will send data through. Adding to that, this approach uses only social relationships, so many adversaries can still launch malicious nodes even when the nodes are authorized. Subsequently, it is important to add other metrics (like PFR or ETX) to defend these compromised nodes.

In cluster-based trust management approach [16], each cluster head needs to send the trust values to the base station that can perform the trust evaluation to all other nodes. This process leads to more energy consumption and could overwhelm the network. Therefore, an approach to improve this proposed solution can let every cluster head evaluates only the trust values of its nodes. This improvement can decrease the number of control messages exchanged between node and reduce the energy consumption. Also, this approach can perform better if it takes into account other trust metrics.



In trust based RPL for the internet of things [17]. The proposed approach has a weak spot which is a node could choose the longest path to send data. Thus, in order to fix this weakness, this approach may use other metrics like hop count to choose shorter path. Moreover, this approach doesn't consider some other attacks like grey/black hole attacks that can still be launched. So, as we previously mentioned, this approach can be improved by adding another metric to detect and avoid this attack.

The main limitation of the proposed approaches in [18,19] is that each node takes decision based on its evaluation without checking others' observations. So, if the node evaluates mistakenly one of its neighbors, it will not use it anymore to send data through. Therefore, in order to improve them, a node needs to calculate the trust values of other nodes especially its parents based on its observation and others' recommendations. Also, these approaches consider only some disruption attacks, so to improve this, these approaches can add other metrics to mitigate the attacks.

The approach proposed in [20] requires that each node needs to have a trust manager to calculate the trust values of all other nodes. Therefore, it has similar weakness as [14], where the nodes require extra hardware and it may have a storage problem as the social approach [15].

## CHAPTER 4

### PROPOSED TRUST AWARE RPL

In this chapter, we present our proposed trust aware routing protocol for low power and lossy network. We first introduce the basic concepts of trust, then present an illustration for our proposed trust management system. At the end, we show how the proposed routing protocol can mitigate many attacks in IoT by detecting and isolating the malicious nodes.

#### **4.1. Trust Model Basic Concepts**

Guaranteeing security is crucial for IoT and needs the creation of a critical trust mechanism. A trust management can provide a solution for many security issues in IoT. Therefore, in the context of IoT, in which the smart things (objects) make decisions, a trust relationship must be set among these things. In this context, trust and security are the most important characteristics required to have safe interactions in any network.

Two types of trust exist: trust in action and trust in recommendation. The trust in action lies on trusting someone based on his/her performance to achieve certain duties; As an example, we can trust a civil engineer to estimate quantities and cost of materials, equipment, or labor to determine project feasibility, but we can't trust a developer for these tasks. However, trust in recommendation lies on the observations of people towards specific person; for example, we can trust a doctor when many people recommend him/her and when we see how s/he behaves (in this case we trust him based on other observations and ours). These two types are valid to be applied in IoT or other networks in order to build safe interactions and communications between nodes. Trust can be computed in several ways, using node's observation and the information gathered from other nodes or monitoring other nodes' behaviors by snooping on others' behaviors.

Our trust aware routing protocol, presented next, combines these 2 types of trust by allowing each node to evaluate the trust of other node based on its behavior and the observations of its neighbors. It takes into consideration many metrics to build trustworthy relationships between nodes including social relationships like *honesty, intimacy and cooperativeness*, and quality of service (QoS) guarantees like packet forwarding rate. It is a reputation-based approach, since the node evaluates the reputation of another node periodically based on its behaviors as seen by the node itself and others.

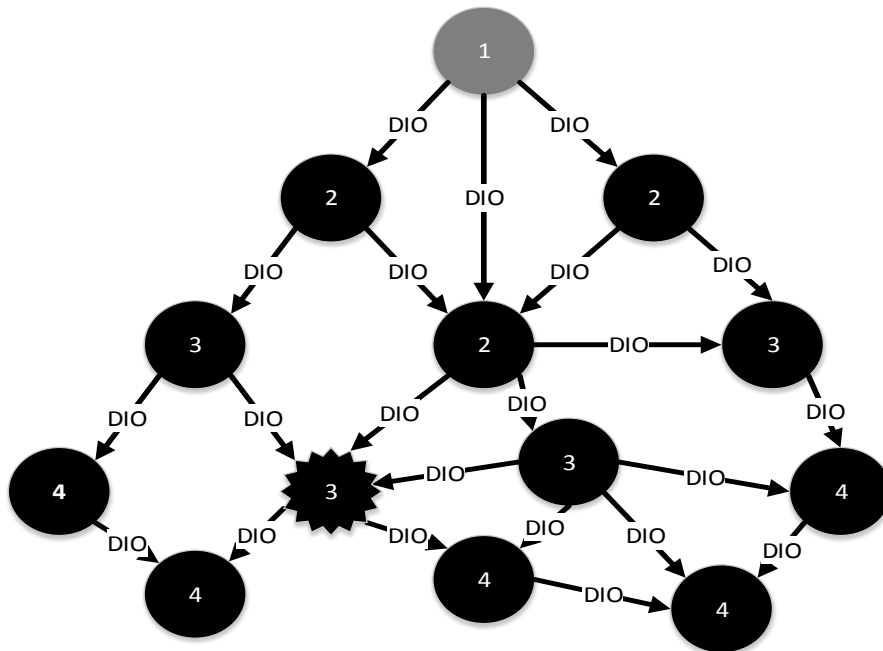
#### **4.2. Proposed Trust-Aware Routing protocol for Low Power and Lossy Network (TARPL-LLN)**

In trust-aware routing protocol for low power and lossy network (TARPL-LLN), the trust will be achieved in 2 concurrent levels based on social and quality of service metrics. At the first level, social relationships will be formed between nodes. This step has a main role in filtering the neighbor nodes of a specific node into social and non-social trusty neighbors. However, at the second level a node will play a snooping role to detect if the social trusty parents are working properly. Using these 2 steps, nodes will be able to calculate another node's reputation. Initially, all the nodes are considered good with an adequate reputation. When the node transfers data through a specific parent, it can update its opinion about that parent by using its observation and other recommendations if exist.

##### **4.2.1. Network Construction**

In TARPL-LLN, the initial construction of the DODAG will be accomplished based on the hop-count metric with no constraints (since all the constraints are satisfied at time zero). The construction starts at the root when it broadcasts DIO messages to its neighbors. Upon receiving a DIO message, the node has to calculate its rank and choose a set of parents. Each node will

calculate its rank by taking the minimum rank of the nodes that sent the DIO messages then add 1 to it. As shown in Figure 4.1, the star node received DIO messages from 3 nodes with ranks equal to 2, 3, and 3, it finds the minimum rank that is equal to 2 and adds 1, hence its rank is 3. If a node  $N$  receives a DIO message, after the calculation of the rank, from a node with rank greater than or equal to its rank, it will not update its rank. When the topology of DODAG is completed, each node will have a set of parents that will include a preferred one. The set of parents for every node  $N$  are the neighbor nodes with ranks less than or equals to  $N$ 's rank. After that, each node needs to select its preferred parent through which it sends the data to the root of DODAG (node with rank 1). Hence, a node will select its preferred parent within 2 levels which are the filtering level that filters the parents into social and non-social trusty parents based on some social constraints (specified below), and QoS level that allows a node to set its preferred parent from the filtered set of parents.



**Figure 4.1: DODAG construction based on hop-count metric, since at time 0 all constraints are satisfied**

#### 4.2.2. First Level: Social trust Relationship (filtering step)

During this level, social trust relationships will be formed between each node and its neighbors based on 3 social metrics which are chosen from a wealth of social metrics that are available [23] “*honesty, intimacy and cooperativeness*”. In TARPL-LLN, we differ from [15], where each node needs to store the trust evaluated values of all other nodes. Each node needs only to store the trust values of its neighbors set.

The 3-social metrics used in TARPL-LLN can be measured as below:

- The *honesty* trust property, it states whether a node is honest or not. In IoT system, a malicious node could be dishonest when providing incorrect services or recommendations. Hence, node  $i$  checks that node  $j$  is honest based on node  $i$ 's direct interaction with node  $j$  at time  $t$ . First, when node  $i$  receives the *honest* value from node  $j$  toward node  $q$ ,  $i$  will check if the percentage difference between  $i$ 's and  $j$ 's evaluation toward node  $q$  is higher than a threshold,  $i$  considers  $j$  as suspicious and thus a zero-honesty value will be for  $j$ .
- *Cooperativeness* trust property, it represents whether the trustee and trustor nodes are socially cooperative [25]. To calculate the cooperativeness of node  $j$  by node  $i$ ,  $i$  keeps a list of its friends which may be modified. Cooperativeness is computed as the ratio of common friends between  $i$  and  $j$ , i.e.  $\frac{friends(i) \cap friends(j)}{friends(i) \cup friends(j)}$ , where  $friends(i)$  denotes the set of friends to the owner of node  $i$ .
- *Intimacy* property, it is the level of interaction experiences which is computed by the number of interactions between nodes  $i$  and  $j$  over the maximum number of interactions between node  $i$  and any neighbor node over the time period  $[0, t]$ .

In our approach, a node calculates the social trust values of its neighbors only. The social trust value of node  $i$  toward node  $j$  can be calculated as given in equation 4.15.

$$T_{ij}(t) = w_1 T_{ij}^{intimacy}(t) + w_2 T_{ij}^{honesty}(t) + w_3 T_{ij}^{cooperativeness}(t) \quad (4.15)$$

This component is calculated based on node  $i$ 's observation and the recommendations of its neighbors. Node  $i$ 's observation takes into account both recent and old observations as given in equation 4.16.

$$T_{ij}^X(t) = (1 - \alpha) T_{ij}^X(t - \Delta t) + \alpha D_{ij}^X(t) \quad (4.16)$$

where  $D_{ij}^X(t)$  is the direct observation of node  $i$  towards node  $j$  at time  $t$ .  $T_{ij}^X(t - \Delta t)$  is the old value of  $i$  towards node  $j$ .  $t$  represents the  $[0 \dots t]$  time.  $\alpha$  is to weigh the direct observation at time  $t$  and old observation at time  $t - \Delta t$

When a node  $i$  interacts with node  $k$  which has a trust value about  $j$ ,  $i$  updates the trust value of  $j$  as in equation 4.17 where  $[0 < \beta < 1]$  is to weigh its observation and other recommendations.

$$T_{ij}(t) = (1 - \beta) T_{ij}(t - \Delta t) + \beta R_{kj}(t) \quad (4.17)$$

When the node calculates this trust value, it can now differentiate between socially and non-socially trusted parents using a shared threshold generated by the root (we can use a threshold equals to 0.5, so the parents that have trust values greater than 0.5 can be good candidates to the second step in our trust management approach).

#### 4.2.3. Second Level: QoS Level (preferred parent selection)

By applying the first trust level, a node can distinguish between trusted and non-trusted parents based on the social relationships built between things. At this level which happens at the same time as the first one (i.e. the node does the 2 levels of trust at the same time), the node needs to know if that trusted parents are working well or not (whether they are forwarding or dropping the data packets). In this case, we introduce a new metric which is the packet forwarding rate (PFR) that is also calculated by the nodes towards their parents. This metric is calculated based on the behavior of the parent through which the data were transmitted. In this

regard, a node sends the packets, then enters a snooping mode to check how many packets were dropped and how many were forwarded. subsequently, a node  $i$  can calculate the PFR of its parent  $j$  as equation 4.18.

$$PFR_{ij} = \frac{FP_{ij}}{FP_{ij}+DP_{ij}} \quad (4.18)$$

where  $FP_{ij}$  is the number of packets node  $j$  has forwarded out of the packets it received from node  $i$  and  $DP_{ij}$  is the number of dropped packets. This value represents the observation of node  $i$  toward node  $j$ , hence, to calculate the final  $PFR_{jfinal}$ , node  $i$  needs to calculate the average  $PFR_{kj}$  coming from nodes  $k$  ( $k$  is any common neighbor for  $i$  and  $j$ ) as given in equation 4.19.

$$PFR_{jfinal} = \frac{PFR_{ij} + \sum_1^m PFR_{kj}}{m+1} \quad (4.19)$$

where  $PFR_{ki}$  is the packet forwarding rate from any node  $k$  towards  $j$  and  $m$  is the number of neighbors that evaluated  $j$ .

$PFR_{jfinal}$  can be combined with other metric which is the remaining energy (RE) in order for the node to choose a *secure* and *short* path to send its data to the border router (route). once the node has the PFR and the other metric, it can calculate the second trust values of the filtered set of parents,  $TR_{ij}$  as given in equation 4.20

$$TR_{ij}(t) = \frac{PFR+RE}{2} \quad (4.20)$$

Where  $TR$  is the trust value of node  $i$  towards node  $j$  that  $i$  can choose to select its preferred parent.

At the end, the node can select its preferred parent from the filtered set of parents, where this parent will have the maximum  $TR_{ij}(t)$ .

### 4.3. Interaction of 2 Neighbor Nodes

TARPL-LLN is an interaction-based approach. Hence, whenever 2 nodes interact, each node needs to update the trust values of its neighbors based on the equations specified in subsection 4.2. Algorithm 4.1 specified how the node behaves when it interacts with its neighbor.

---

**Algorithm 4. 1: When node  $i$  receives DIO message from node  $j$**

---

1. *If*  $rank_i > rank_j$ :
2.      $set\_of\_parent.add(j)$
3.  $Nodes\_list = DIO.get\_nodes\_list()$
4. *Foreach*(node  $k$  in  $Nodes\_list$ ):
5.     *If*  $k$  belongs to  $set\_of\_neighbors$ :
6.          $T_{ik}(t) = (1 - \beta)T_{ik}(t - \Delta t) + \beta R_{jk}(t)$
7.     *If* node  $k$  belongs to  $set\_of\_parents$ :
8.          $PFR_{ik} = \frac{PFR_{ij} + PFR_{ik}}{2}$
9.  $D_{ij}^{intimacy}(t) = \dots$
10.  $D_{ij}^{honesty}(t) = \dots$
11.  $D_{ij}^{cooperativeness}(t) = \dots$
12.  $T_{ij}^{intimacy}(t) = (1 - \alpha)T_{ij}^{intimacy}(t - \Delta t) + \alpha D_{ij}^{intimacy}(t)$
13.  $T_{ij}^{honesty}(t) = (1 - \alpha)T_{ij}^{honesty}(t - \Delta t) + \alpha D_{ij}^{honesty}(t)$
14.  $T_{ij}^{cooperativeness}(t) = (1 - \alpha)T_{ij}^{cooperativeness}(t - \Delta t) + \alpha D_{ij}^{cooperativeness}(t)$
15.  $T_{ij}(t) = w_1 T_{ij}^{intimacy}(t) + w_2 T_{ij}^{honesty}(t) + w_3 T_{ij}^{cooperativeness}(t)$
16. *If*  $rank_i > rank_j$  and all the trust values are satisfied:
17.      $rank_i = rank_j + 1$
18. *Else*:
19.     *Remove*  $j$  from the set of parents
20.  $i$  sets its rank, RE and trust values then broadcasts DIO messages

In this algorithm node  $i$  receives a DIO message from node  $j$  containing trust values for node's  $j$  neighbors. First, node  $i$  checks if the rank of node  $j$  less than its, in this case node  $j$  could be a parent node for  $i$ . So, it adds  $j$  to its parents set. Then, it checks every node  $k$  in the set of  $j$ 's neighbors. Node  $i$  updates the trust value for every node  $k$  that is common neighbor for  $i$  and  $j$  based on the equation  $T_{ik}(t) = (1 - \beta)T_{ik}(t - \Delta t) + \beta R_{jk}(t)$ . Also, it updates the trust values

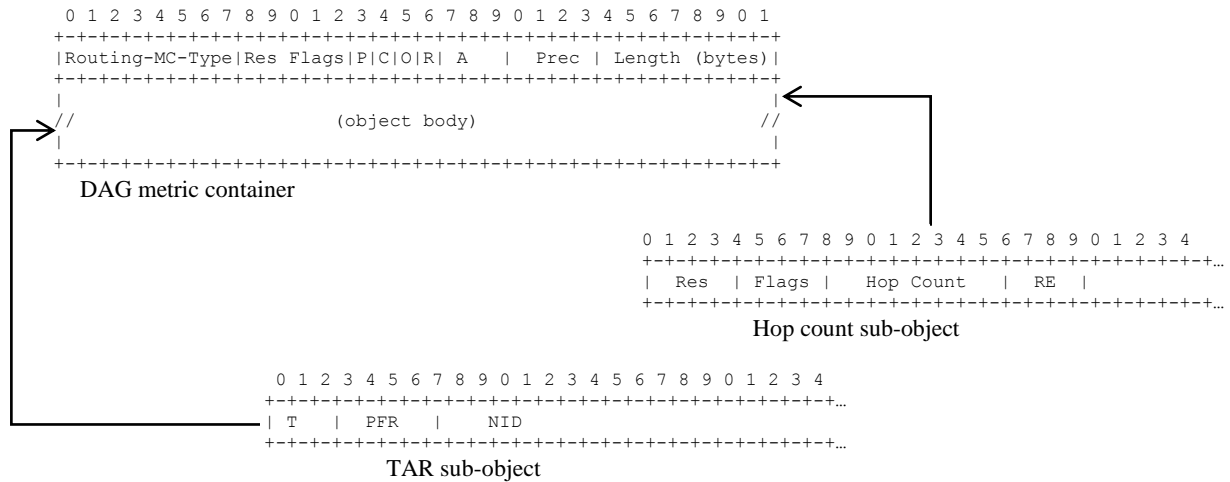
for every node  $k$  that is a parent for  $i$  and neighbor for node  $j$  based in the equation  $PFR_{ik} = \frac{PFR_{ij} + PFR_{ik}}{2}$ . Second, node  $i$  updates all the social trust values for node  $j$  based on the



current observation and the previous observation. Finally, node  $i$  checks if all the trust values of node  $j$  are satisfied (i.e. based on the threshold) and the rank of node  $j$  is less than its rank so node  $j$  will be set as parent for node  $i$ . After these steps node  $i$  updates the received DIO message and broadcast it to its neighbors.

#### **4.4. Trust representation in RPL DIO message:**

This section specifies how the trust metrics and constraints will be carried in the DIO message in order to let the node find its trusted and shortest path towards the route. The metrics and constraints are carried in objects (objective code point-OCP) that are optional from the point of view of an RPL implementation. To implement the trust aware routing protocol for RPL, we introduce a trust aware routing (TAR) object in the DAG metric container of the DIO message [32], [34]. As shown in Figure 4.2, the TAR object contains sub-objects that correspond to the evaluations of nodes except the first one which is used to calculate the rank of a node and its remaining energy. Our approach uses TAR object as a constraint where this can be specified by the  $C$  flag on the DAG metric container when the border router broadcasts DIO messages. Each node that participates in constructing the DODAG, updates the DIO messages by inserting its TAR sub-objects that convey the trust values of its neighbors according to equations 4.15, 4.16 and 4.18.



**Figure 4.2: TAR sub-objects within the DIO DAG Metric Container**

The fields specified in TAR sub-objects are:

1. NID: It is the node id, representing the identifier of the evaluated neighbor  $j$ . It can be an IPv6 address.
2. T: It represents the social trust value of node  $j$  evaluated by the node that is broadcasting DIO messages.
3. PFR: It is the packet forwarding rate for node  $j$  calculated according to equation 4.18.
4. RE: It represents the remaining energy percentage of the node that has this NID.

Note that there is also a sub-object representing the information of the node itself that contains only its remaining energy. Upon receiving a DIO message, the node updates its trust values for its neighbors based on the equations stated above, then it updates the DIO message by replacing the old sub-objects by its sub-objects that contain the new trust values evaluated by the node itself.

#### 4.5. Illustrated Examples:

In TARPL-LLN, many attacks can be detected and avoided by finding the malicious nodes and isolate them from an IoT network. The subsections below show scenarios about the attacks that can be avoided using the proposed approach.

##### 4.5.1. Self-promoting attack

In this attack, a malicious node will promote its significance by showing good recommendations for itself. Hence, other node with rank greater than its' will select it as a preferred parent as shown in Figure 4.3 where the malicious node (MN) attracts other nodes to send their data through it. Our trust protocol can mitigate this attack by using the honesty constraint at its first level. Hence, a node will be evaluated by other nodes based on its honesty and other metrics.

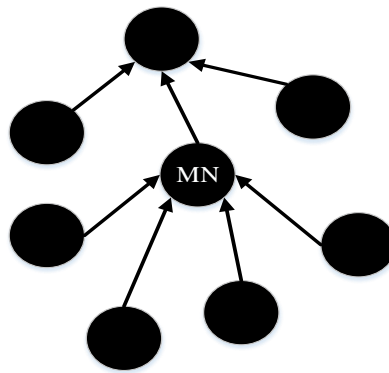


Figure 4.3: Self-Promoting Attack

##### 4.5.2. White-Washing attack

A malicious node could disappear and reenter the network to wash away its bad reputation that was evaluated by other nodes. Therefore, a new good reputation will be assigned to that node where it can participate in the network. Our trust-based approach can detect this

attack by allowing a node to memorize the trust evaluations of its neighbors. Therefore, if a node tries to silt its bad reputation by using white-washing attack without moving to other place (stay in the same community), TARPL-LLN can detect this attack and prevent it from participating in the network. In addition, if the node goes to other place, it cannot participate directly, because at the first level, our approach uses the intimacy and cooperativeness constraints, hence a new node has a slight chance to cooperate in the network.

#### **4.5.3. Bad-Mouthing Attack**

In this type of attacks, a malicious node can destruct the reputation of a well-behaved node by broadcasting bad recommendations, so as to decrease the chance of this good node being selected as a preferred parent. In TARPL-LLN, the honesty and cooperativeness constraints at the first level limit the ability of bad-mouthing attack to launch IoT network, since a node that provides bad reputation about a good node will be judged as a dishonest node.

#### **4.5.4. Ballot-Stuffing Attack**

A malicious node can boost the reputation of another bad node by providing a good reputation for it in order to increase its chance to be chosen as preferred parent for some nodes. This is the case of ballot-stuffing attack. This attack can be alleviated in TARPL-LLN, since the trust relationships, at the first level, are formulated based on honesty, intimacy and cooperativeness constraints. Therefore, if a node boosts the reputation of another bad node, it will be classified as dishonest node, and the value of its intimacy will be smaller.

#### **4.5.5. Sinkhole Attack**

This attack is similar to good-mouthing attack; hence, it can be avoided during the first level since when a node provides good reputation about a bad-behaved node, it will be judged as a dishonest node and further communication with it will be avoided.

#### **4.5.6. Hello Flooding Attack**

For joining the network, the node needs to broadcast initial message as hello message (DIS message). In this case, an attacker can introduce itself as neighbor node to many nodes by broadcasting DIS messages with robust routing metrics and participate in the network. this attack can be mitigated in TARPL-LLN since there are the cooperativeness and intimacy constraints that will isolate the attack and prevent this attack from being executed.

#### **4.5.7. Rank Attack**

In this attack, the malicious node will change its rank in the sake of attracting other nodes for selecting it as preferred parent. This attack has no effect on our protocol, since the selecting of the preferred parent is based on the constraints in the 2 levels which are social relationships and quality of service. Therefore, even if the node does modify its rank, this doesn't mean that other node will prefer it to send the data through.

#### **4.5.8. Selective Forwarding Attack**

This attack occurs when a malicious node selects some packets to forward and others to drop. It could lead to Denial of Service (DoS) attack that may disrupt routing paths. Therefore, this attack takes place by selective forwarding packets. In TARPL-LLN, the selective forwarding

attack will be avoided, since any node at the second level calculates the packet forwarding rate (PFR) of its parents, then it chooses the one that has the highest PFR as preferred parent. In this case the node that drops some of the packets will have lower PFR and as a result it will not be chosen as preferred parent for any node.

#### **4.5.9. Blackhole Attack**

Blackhole attack is a special case of selective forwarding attack, where in this attack the malicious node will drop all the data packets sent by their children. This attack also can be mitigated and avoided at the second level, where the malicious node that is dropping all data packets will have a  $PFR = 0$ , therefore sending the data through this adversary node in the future will be impossible.

As we already mentioned, a malicious node can also perform other attacks like Sybil and identity attack where a node clones other node's identity. In our approach, we assume that these attacks can be handled by an authentication system that doesn't allow a node to mimic another node identity.

#### **4.6. How TARPL-LLN differs from other works**

The aim of our approach is to detect and isolate the malicious nodes from the IoT network. In order to innovate a new approach, we surveyed works found in the literature as shown in Chapter 3 TARPL-LLN differs from others by the following:

1. No need to use any extra chip like trust platform module (TPM), since the node will be calculating the trust values of its neighbors based on the information gathered from the received DIO messages.

2. Each node will need just to store information about its neighbors, so in this case, the storage constraint LLN will not be impacted.
3. Each node will receive DIO messages every trickle time, where it can update the trust values dynamically (reputation-based), thus the evaluation of other nodes from a specific node will be always updated.
4. TARPL-LLN can be applied to cluster and non-cluster-based networks
5. Proposed approach is not centralized trust system, every node stores trust values for its neighbors and it takes decision based on the trust values that were calculated according to its observation and others' recommendations.
6. There are two trust levels which are social and QoS levels, this division gives more efficiency for detecting the malicious node, and to consume less energy since the second level will be omitted when the trust value in the first level is less than fifty percent.
7. In TARPL-LLN, the node will not only choose the most trusted path, it also chooses the one that has the highest remaining energy (in this case, the consumption of energy will be balanced) and the one that is the shortest where the construction of the tree is based on the rank of the node.

## CHAPTER 5

### PERFORMANCE EVALUATION

In this chapter, we evaluate the performance of the proposed RPL based trust aware routing protocol. Our experiments are performed using the simulator Contiki 2.7 while integrating the proposed trust aware routing protocol into RPL. We use some standard performance metrics namely remaining energy, packet latency, packet delivery ratio (PDR), throughput, percentage of malicious detection, and packet loss ratio to evaluate the performance of our approach.

#### 5.1. The Contiki Simulator

Contiki is an open-source multitasking operating system (OS) that is organized for IoT applications. Contiki OS is the state-of-the-art OS established for tiny networked embedded systems. It focuses on tiny low-power network embedded microcontrollers. Processes in Contiki are seen as event handler, which handle events thereby, making it possible for the kernel, applications and drivers to interact and function sufficiently. This benefits in executing several processes concurrently [36]. We chose Contiki OS, because of its optimized uIPv6 stack, which contains 6LowPAN as an adaptation layer to support routing over the link and network layer. It implements a trivial network stack called *Rime*, which provides protocols for data collection and route discovery to a destination node. The socket-like API includes application function to support uIPv6 stack. UDP, TCP and the ICMP are the transport layer communication protocol that supported Contiki to be used for sending control messages from the IPv6 LoWPAN to the network layer through the Socket-like API function. Communication over the radio is achieved using *Rime* stack. The *Rime* contains network protocol library which includes various low-level



primitives. It also takes care of the medium access control. The platform layer handles hardware low level abstraction and the responsibility of porting the CPU to the hardware drivers.

Contiki OS implements different module for a specific task. The `contiki/core/net/` module comprises specific folder that implements different layers of the protocol stack, which includes the MAC, *Rime* and RPL. Some of the vital files in `rpl` folder are *rpl.c*, *rpl-dag.c*, *mrhof.c*, *Of0.c*, *icmp6.c*. The *rpl.c* file implements the ipv6 Routing Protocol for Low-Power and Lossy Networks (RPL). The *rpl-dag.c* file contains logic implementation use in constructing Directed Acyclic Graph (DAG)in RPL. *mrhof.c* contains the implementation of minimum rank hysteresis objective function, which uses ETX as a routing metric. *of0.c* implements objective function zero which uses hop count as a routing metric and finally the *icmp6.c* file contains functionalities for RPL controls message that has the information about the parameters use for routing information. Contiki also features a cross layer network simulation with COOJA which is a simulator environment that is capable of simulating different hardware platforms, including sky mote used in this work.

## 5.2. Simulation, Metrics and Parameters

In this section, we have integrated our trust aware strategy for low power and lossy network with RPL protocol, hence, we can study the performance of TARPL-LLN using some standard performance metrics which are specified below:

- **Remaining Energy:** it is a remaining energy estimation for the whole IoT network which is calculated based on each node remaining energy. In our study, we take the average percent of consumed energy on time for all the nodes in the whole network setup.

- **Packet Latency:** The Latency is defined as the amount of time taken by a packet from node to reach the sink node. Packet latency for the whole network is the average of the latencies of all the packets in the network from all the nodes.
- **Packet Delivery Ration (PDR):** The number of received packets at the sink to the number of sent packets. We take the average PDR of all the packets received successfully at sink.
- **Throughput:** It is the amount of data moved from a node to the sink node in a given time period. In our analysis, we calculate the average throughput for all nodes.
- **Percentage of Malicious Detection:** It is the number of detected malicious nodes to the number of malicious nodes.
- **Packet Loss Ratio:** It is the number of packets sent by the node minus the number of packets reached the sink to the number of sent packets. In this study, we calculate the packet loss ratio of the network by calculating the average packet loss ratios for all nodes.

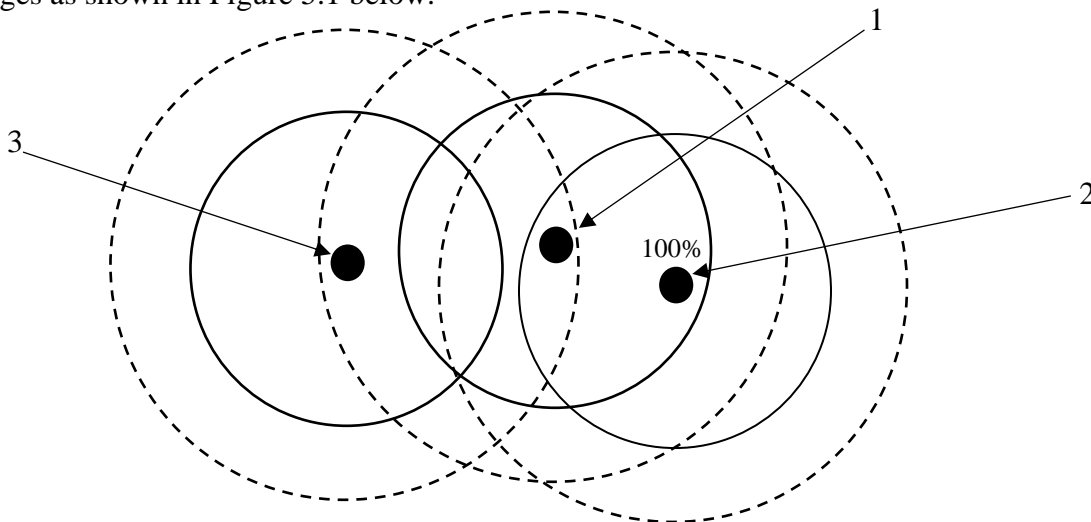
### 5.2.1. Phases of Evaluation and Network Setup:

Because RPL uses two different objective functions and many metrics to build its topology (DOGAG), we divided our experiments into two phases. In the first phase of the simulations, we study the performance of OF0, MRHOF and TARPL-LLN in terms of three metrics: Remaining Energy in the network, network latency and packet delivery ratio in order to make sure that our approach doesn't impact the performance of RPL protocol. In the second phase, we evaluate our approach in terms of the metrics stated above, then compare it with other approaches and the standard RPL protocol.

Simulating losses in wireless medium is very important because it illustrates the actual environment where the sensor nodes are deployed. The more accurate the simulations of the radio medium the closer are the results to the actual radio medium. Therefore, in this subsection, we explain how we simulate the radio medium in COOJA followed by the network setup used for our simulation. Then, we describe how we calculate the chosen performance metrics from the collected data and finally running the simulations in an organized manner.

### 5.2.1.1. Failing in Transmission

The link failure model is simulated by the use of unit disk graph model (UDGM) in COOJA [37]. It uses two different range parameters which are transmission and interference ranges as shown in Figure 5.1 below.



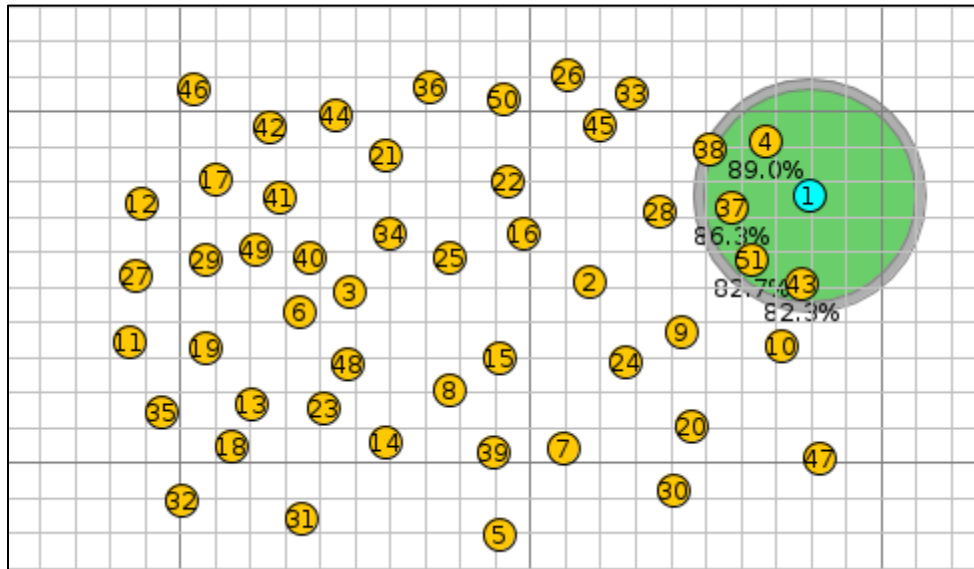
**Figure 5.1** The small circle around node 1 denotes the transmission range (R) while the bigger circle denotes its collision with other radios. The figure shows the reception ratio of the transmission between node 1 and 2. The node 3 is inside the collision range of node 1.

The radio ranges (transmission and interference) increase according to radio power. Also, a node has transmission and reception ratios which also can be configured and organized using UDGM. Since in this work, we are interested to study the detection of malicious nodes, we keep the transmission ratio to 100% and let the receiving ratio depend on the distance between the nodes.

Indeed, the probability of a packet to successfully reach another node increases as the distance between them increases and are in transmission range of each other.

### 5.2.1.2. Network Setup

To perform the intended simulations, we design a specific network in the COOJA simulator containing several client nodes and one server node representing the root of the DODAG. The network topology is shown in Figure 5.2, where the server uses a sample application `udp-server.c` and the client nodes use `udp-client.c`.



**Figure 5.2** one sample of RPL Network setup that contains 50 clients and 1 server that acts as a sink node in COOJA simulator. The node with id 1 is the sink node and the others are the client nodes that send data to the server.

After integrating our trust aware routing protocol into RPL, we are able to start the experiments needed to analyze and study the specified parameters. The various parameters for simulation and its environment are listed in Table 5.1. Indeed, the start delay is 50s which is the initial time for the nodes to start transmitting messages to the sink node. This initial time is an approximation for the time sufficient to ensure that the packet sent to the server will not get lost because of the lack of network connectivity.

<b>Simulation Parameters</b>	<b>Values</b>
Start Delay	50s
Simulation tool	Contiki/COOJA/2.7
Mote Type	Sky Mote
Simulation Time	1 hour
Simulation Cover Area	300m x 300m
RPL MOP	NO_DOWNWARD_ROUTE
OF	OF0, MRHOF, our approach
Send Interval	4s
TX Ratio	100%
TX Range	50m
Total Number of Nodes	10...130
percentage of malicious nodes	10...30%

**Table 5.1: Network related parameters used in simulation analysis**

The send interval represents the interval time between two successive messages sent from a node. Both the start and send intervals have been added a small random number. Then, the rate of transmission packet per a node is  $1 \text{ packet}/(\text{Send Interval} \pm \text{Random Number})$ . Also, the minimum number of packets sent can be calculated as in equation 5.21

$$1 \text{ packet}/(\text{Send Interval} + \text{Random Number}) * \text{Simulation Time} \quad (4.21)$$

and the maximum number of packets sent is calculated as equation 4.22

$$1 \text{ packet}/(\text{Send Interval} - \text{Random Number}) * \text{Simulation Time} \quad (4.22)$$

Indeed, the packet transmission starts after the Start Delay (50s), hence the actual simulation time will be less by Start Delay (Simulation Time – Start Delay). Since each sensor node will get different random number when it starts sending packets, the correct number of packets sent

cannot be pre-computed and that is why we need to measure the number of packets sent precisely in real time. This can enable the fair computation of packet delivery ratios and packet loss ratio. In addition, we see in the table that we set the RPL mode of operation to *No Downward* routes, this is because we are interested in using the multipoint to point traffic for this evaluation and to limit the scope of our study.

### 5.2.1.3. Calculating the performance metrics

In the generated network, each node N sends a UDP message with body “Hello N” (N represents the node id) to the server after each *send interval time*. The client node prints a message ‘Hello N sent to Server’ as soon as it sends a packet, similarly, Server prints a message ‘Hello N received from Client’. Therefore, this allows us to find the receiving time of the packet at server and hence calculate the Latency for all the packets. The latencies for all node are used to compute the network latency as shown in the equation 4.23.

$$Network\ Latency = \sum_{i=1}^n (RecvTime(i) - SentTime(i)) / n \quad (4.23)$$

Where N is the total number of packets received successfully and *i* represents the id of the node that sent the packet.

To measure the average packet delivery ration (PDR), we calculate the number of packets sent by all the nodes to the server and divide that number by the number of successfully received packets as shown in the equation 4.24 below:

$$PDR = (Total\ packet\ received / total\ number\ sent) * 100 \quad (4.24)$$

To study the power consumption metric, we use the mechanism of Powertrace system available in Contiki [38] [39]. Powertrace is a system for network-level power profiling for low-power wireless networks which estimates the remaining energy for CPU processing, packet transmission and listening. This mechanism can be used to calculate the percentage of radio on

time and then we can compute the power consumption for all the network based on the remaining energy of each node.

In order to calculate the throughput in the network, we measure the number of packets received at the sink node, then divide this number to a specific time. This time is defined, when the client nodes start sending packets to the server. Therefore, the throughput in our network is defined as the number of packets received at the sink node within an amount of time.

Regarding the packet loss ratio, its measurement can be achieved by calculating the number of lost packets to the time specified when the network is created. The number of lost packets can be calculated by subtracting the number of packets received at the sink node from the number of packets sent by the client nodes.

Finally, to be able to measure the percentage of malicious detection, we make 10% of the client nodes in the network to be malicious nodes. These malicious nodes could behave as any type of attacks specified in section 2.3. They are distinguished by a new type in COOJA, so to find the percentage of isolated malicious nodes, we check how many attacks were detected and divide this number by 10% of the number of nodes exist in the network.

### **5.3. Required Numbers of Runs**

To ensure credible simulation results, use central limit theorem to estimate the number of runs needed for every experiment while achieving at least a 90% confidence level [40]. First, we did 5 different runs with different seeds for each simulation metric. The results for the 5 runs for the delay are given in Table 5.2.

Then we calculated the standard deviation using the following formula:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{N}}$$

Where  $\sigma$  is the standard deviation,  $x_i$  is the reading at run  $i$ ,  $\mu$  is the mean and  $n$  is the number of runs which is equal to 5 in our experiments.

The +/- precision value used is 5%. The central limit theorem formula used to get the required number of runs is:

$$n = \left( \frac{100 \times z \times s}{r} \times \mu \right)^2$$

where  $n$  is the number of runs,  $\mu$  is the sample mean,  $r$  is the precision level,  $z$  is the normal variate, which is 1.645 constant for 90% confidence interval, and  $s$  is the standard deviation.

**Table 5.2: Number of runs for request delay**

<b>Number of runs for calculating delay</b>			
<b>Run</b>	<b>OFO</b>	<b>MRHOF</b>	<b>Our Approach</b>
1	1.419715	1.04568	1.0532
2	1.32557	0.9898	1.004
3	1.4280801	1.0684	0.971
4	1.3615	1.012	0.992
5	1.4018827	0.967	0.975
<b>Mean:</b>			
	1.38734956	1.016576	0.99904
<b>Standard deviation:</b>			
	0.043019253	0.041013545	0.033059159
<b>z:</b>			
	1.654	1.654	1.654
<b>r:</b>			
	5	5	5
<b>number of runs:</b>			
	3.897880626	1.902240073	1.193659231

#### **5.4. Phase 1: Performance of OF0, MRHOF and TARPL-LLN with no attacks**

This phase is used as a baseline with which we are going to compare our approach. In this phase, we have performed three simulations in order to prove that TARPL-LLN has similar or better performance when there are no malicious (selfish) nodes. Hence, we investigate the performance of our approach comparing it with RPL standard protocol and more specifically

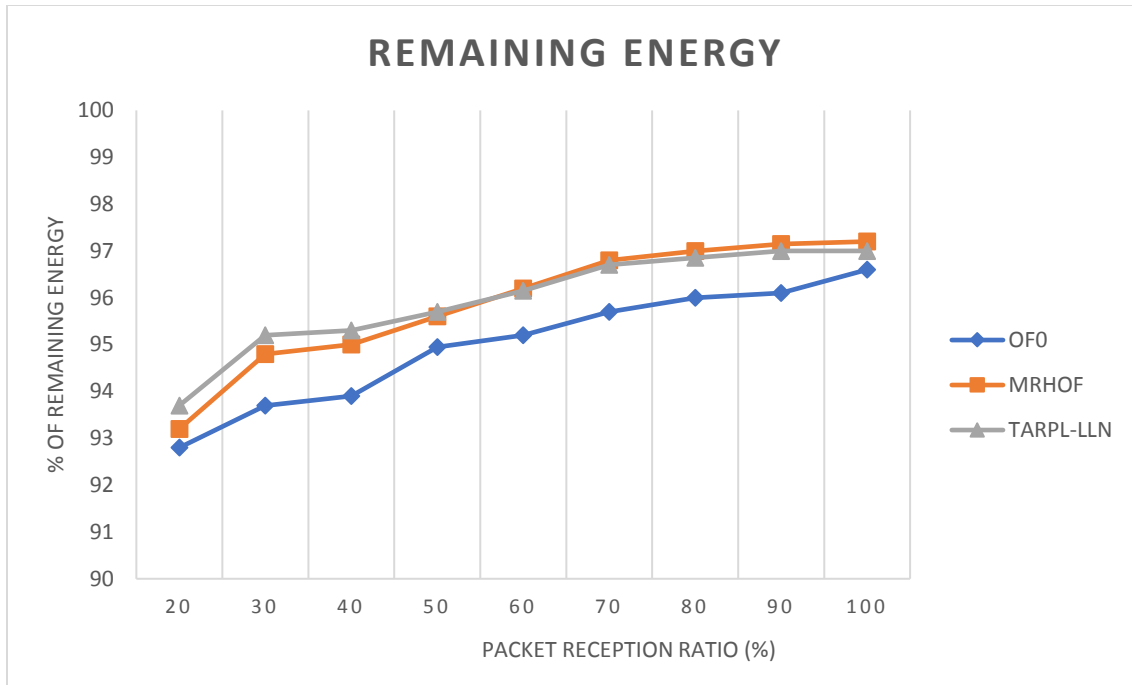


with RPL based on the objective function zero and the minimum rank with hysteresis objective function (MRHOF).

The objective of the experiments in this phase is to evaluate the two objective functions OF0 and MRHOF from one side and TARPL-LLN from the other side in terms of remaining energy, latency and packet delivery ratio. Therefore, we can make sure that our approach will not have negative impacts on the network when there is no attack or malicious node. In order to do these experiments, we set the number of nodes to hundred and the simulation time to one hour, where the receiving ratio varies from twenty to hundred percent.

#### **5.4.1. Remaining Energy in the Network**

We measure here the percentage of energy remained in the network while varying the percentage of packet reception ratio (PRR is defined as a percentage of nodes that successfully receive a packet from the tagged node among the receivers that are within transmission range of the sender at the moment that the packet is sent). As Figure 5.3 shows, the consumption of the energy in the network is high when PRR is low. Whenever PRR increases, the remained energy in the network is high where this is expected because when the PRR is low, the network will be lossy. This leads to have more packet retransmissions than a lossless network.



**Figure 5.3: Energy Consumption for OF0, MRHOF and TARPL-LLN.**

The objective function zero OF0 selects the shortest path based on the number of hops, without taking into consideration any other metric, and that is why we can see in the graph that OF0 has more energy consumption (i.e. less remaining energy). But this is not the case with MRHOF, which outperformed OF0, since it uses the rank quality to choose the path from the node towards the sink. This rank quality depends on some metrics specified in subsection 2.2.4. Finally, we see that TARPL-LLN consumes less or same energy as MRHOF, hence we are sure that our approach has no impact on the energy consumption of the network even when there are no malicious nodes. TARPL-LLN has better performance than OF0 since the nodes with low PRR will be assumed as selfish nodes so no packet will be sent through them. Hence this leads to less packets' retransmissions in the network and consequently less consumption of energy. In the next section, we compare the remaining energy in TARPL-LLN with MRHOF when some of the nodes in the network are malicious.

### 5.4.2. Packet Delivery Ratio

The graph below in Figure 5.4 measures the packet delivery ratio to the sink node varying the percentage of packet reception ratio. It shows an increase in packet delivery ratio whenever the packet reception ratio increases. As seen, TARPL-LLN has better results with low PRR and becomes almost the same as MRHOF when PRR gets higher percentage.

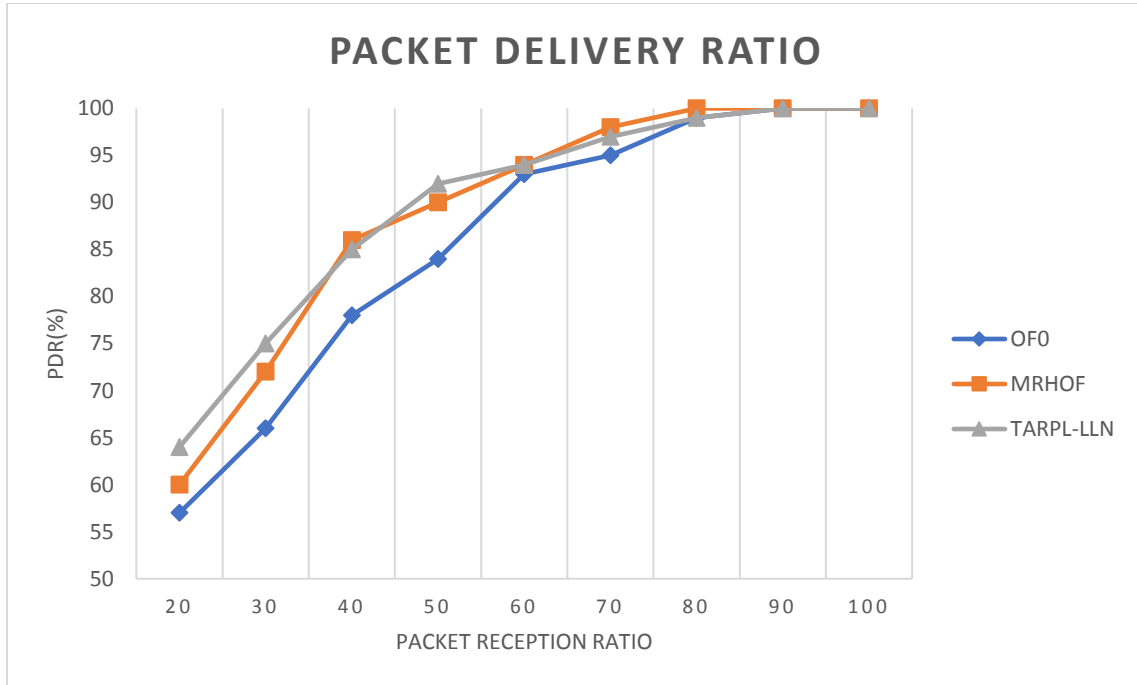
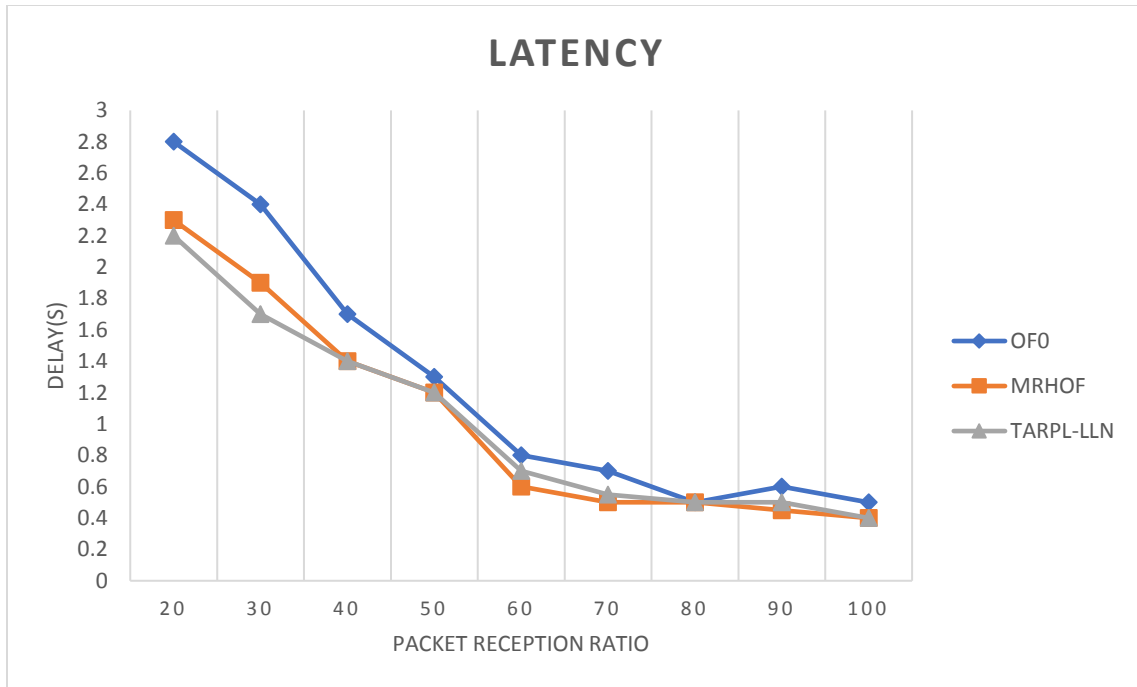


Figure 5.4: Packet Delivery Ratio of OF0, MRHOF and our approach

### 5.4.3. Latency

In this graph (Figure 5.5), we measure the latency in the network varying the packet delivery ratio. As seen, our approach has low delay than MRHOF and OF0 when the packet reception ratio is small. This ensures us that even without any malicious node, the delay in the network will decrease after the usage of TARPL-LLN.



**Figure 5.5: Latency for OF0, MRHOF, and our approach.**

#### 5.4.4. Phase 1 outcomes

Based on the experiments that were investigated in this phase, we can make sure that our approach will not affect the network negatively when no malicious nodes exist. Also, the experiments show that our approach has better performance when the network is more lossy, since the nodes that have low packet reception ratios will be assumed as malicious nodes.

#### 5.5. Phase 2: Performance of MRHOF and our Approach with the Presence of Malicious Nodes

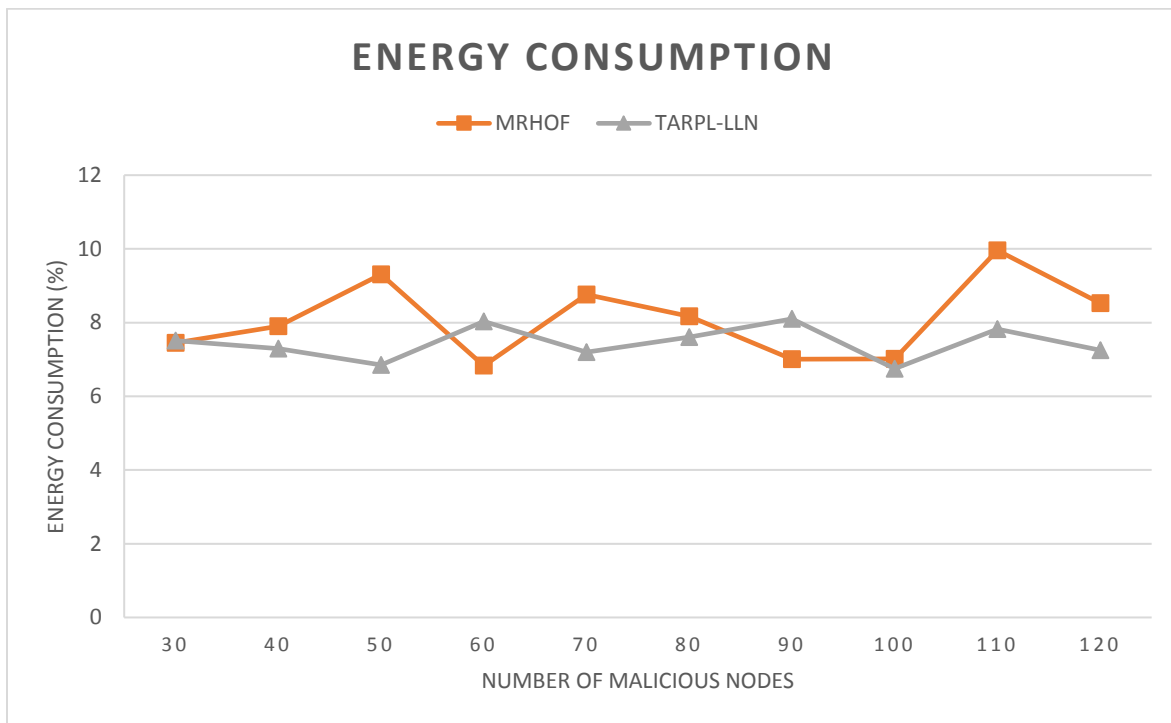
As noticed in the previous simulations, MRHOF and TARPL-LLN are more efficient and have better performance than OF0. Hence, in this phase, we did the comparison between our approach and MRHOF with the presence of some malicious nodes. In the next subsection, we compare our approach with 2 related approaches which are “trust-based service management for

social IoT system” (social) and “using trust management to defend against routing distribution attacks” (PFI) approaches.

### 5.5.1. Performance with 20% Malicious Nodes

In this subsection, we measure the percentage of consumed energy, delay, and throughput while varying the number of nodes within a network composed of one single DODAG, where 20 percent of the nodes are malicious (the malicious behavior is random).

First, Figure 5.6 shows the distribution of the consumed energy between the nodes in the network.

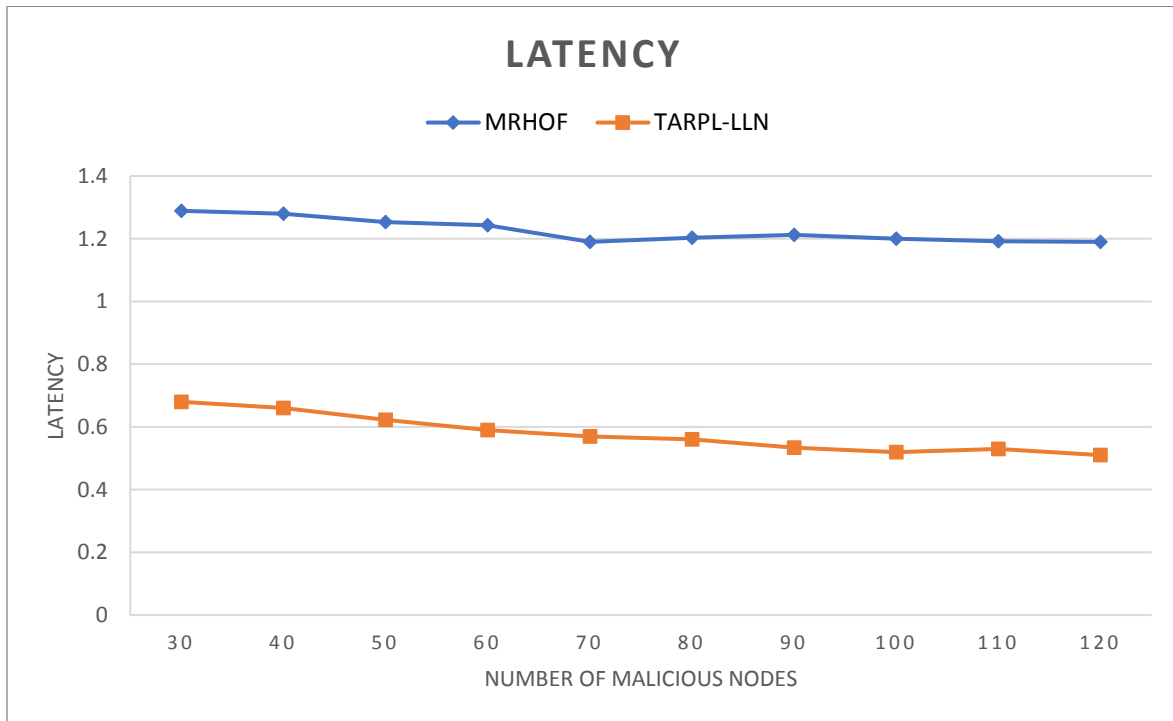


**Figure 5.6: Energy Consumption of TARPL-LLN and MRHOF objective function.**

As seen, the consumption of energy in TARPL-LLN is more balanced than MRHOF (there is a fluctuation in the consumption of energy) since the node uses the remaining energy metric while finding the best path to send the data to the sink node. The simulation results expose that in our approach 68% of nodes have a remaining energy more than 85% and the rest between 70% and

80%, while in MRHOF 21% of the nodes have remaining energy less than 70% where this can affect the network lifetime over time.

Second, we measure the delay in the network while varying the number of nodes from 30 to 120 nodes, Figure 5.7 shows the delay in the network with 20% of malicious nodes.



**Figure 5.7: latency of TARPL-LLN and MRHOF objective function.**

In Figure 5.7, we clearly see that in TARPL-LLN, the latency stays always between 0.5 s and 0.7 s, however, in MRHOF, the latency is always greater than 1.2s. This proves that our approach has the capability to detect and isolate many malicious nodes without being impacted by the number of nodes in the network.

Last but not least, we measure the throughput in the network while varying the number of nodes from 30 to 120 nodes, Figure 5.8 shows the delay in the network with 20% of malicious nodes. Figure 5.8 also shows that the throughput in TARPL-LLN has higher performance in all cases since the size of the network doesn't impact our approach.

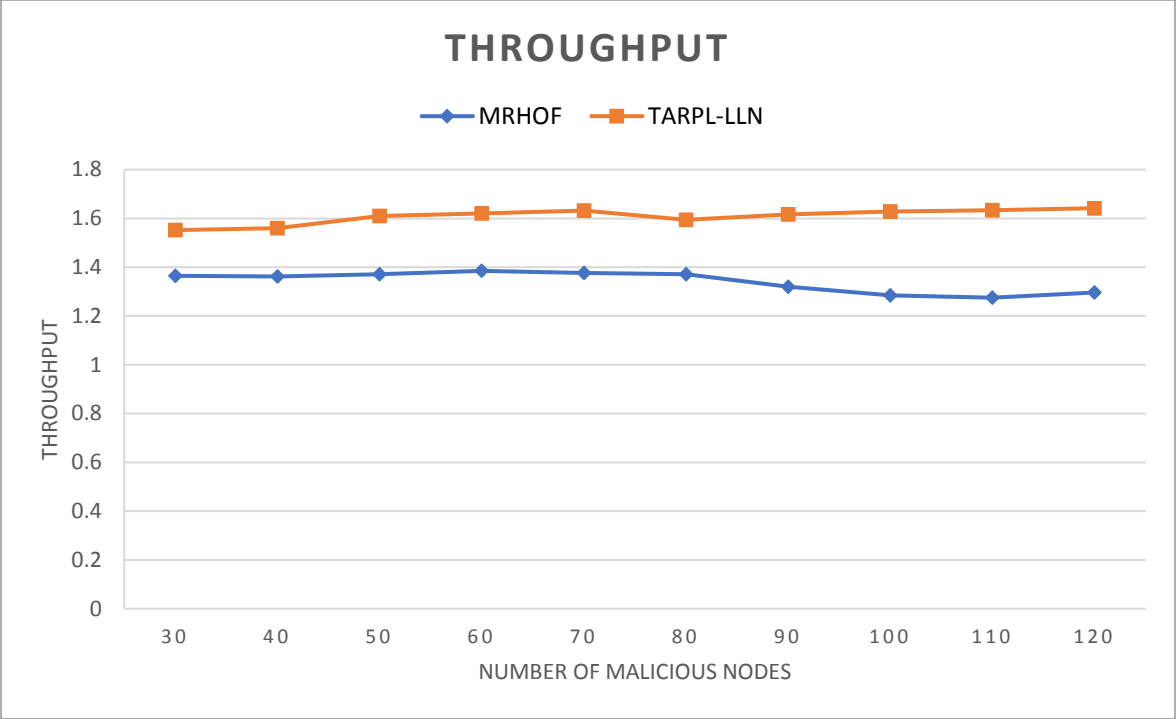
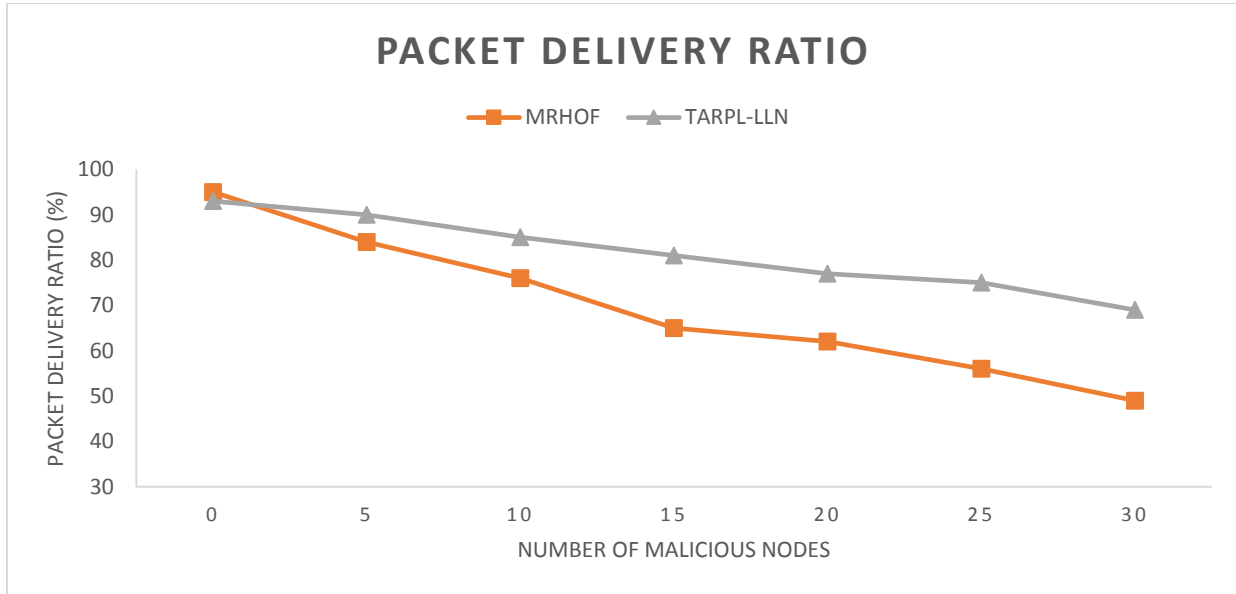


Figure 5.7: latency of TARPL-LLN and MRHOF objective function.

### 5.5.2. Packet Delivery Ratio

In this comparison, we measure the packet delivery ratio for a network consisted of hundred nodes while varying the percentage of malicious nodes from zero to 30. Figure 5.9 illustrates the variation of packet delivery ratio in the network in the presence of malicious nodes.



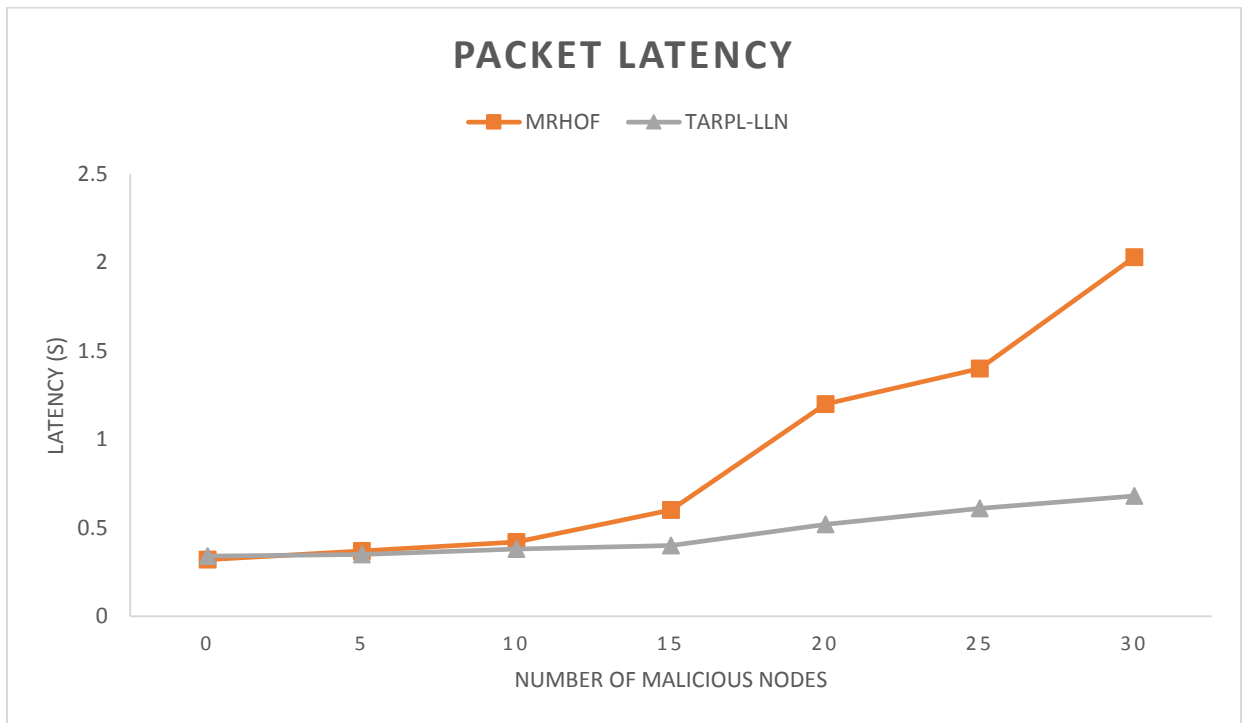
**Figure 5.9: Packet Delivery Ratio of our approach and MRHOF objective function.**

This simulation result shows that TARPL-LLN performs better than RPL standard routing protocol in terms of packet delivery ratio (PDR) when there are malicious nodes. We notice that as the number of malicious nodes increase, PDR decreases roughly in MRHOF where it reaches 49% when 30% of the nodes are malicious. However, in TARPL-LLN there is a decline in PDR, but it is slightly small since many attacks will be isolated.



### 5.5.3. Packet Latency

Here, we measure the latency in a network consisted of hundred nodes while varying the percentage of malicious nodes from 0 to 30 percent. The average latency results are included in Figure 5.10, where it shows that from zero to seven misbehaving nodes in the network, MRHOF and TARPL-LLN lead to almost equal average latency. However, as the number of malicious nodes increase, we can clearly see that there is a severe increase in the latency in MRHOF curve.

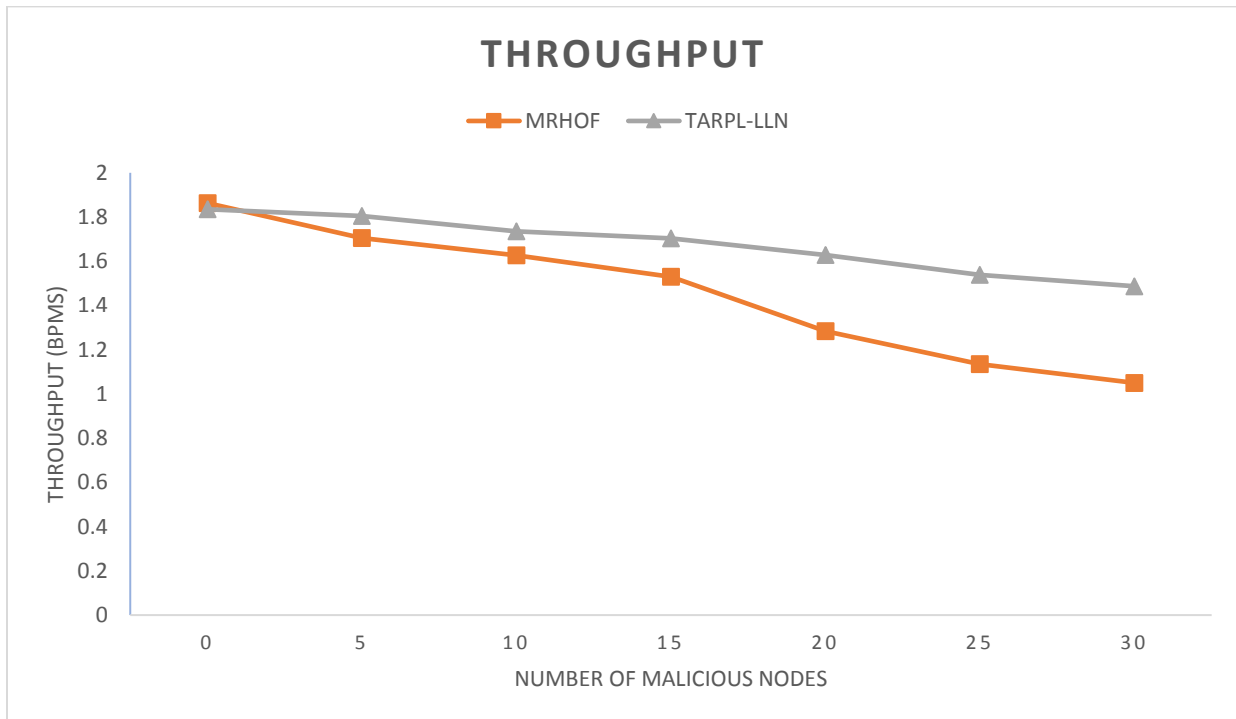


**Figure 5.10: Packet Latency of our approach and MRHOF objective function.**

The results in this experience reveals that our approach performs well and better than the standard protocol in the case of existing malicious nodes in the network. This is due to the ability of detecting the misbehaved nodes and removing them from the DODAG in a quick way.

#### 5.5.4. Throughput

In this experience, since the interval time to send packets is big which is 4 seconds, we get small values for throughput which is measured in bits per seconds (bps). Figure 5.11 presents the throughput in the network while modifying the percentage of malicious nodes from 0 to 30.

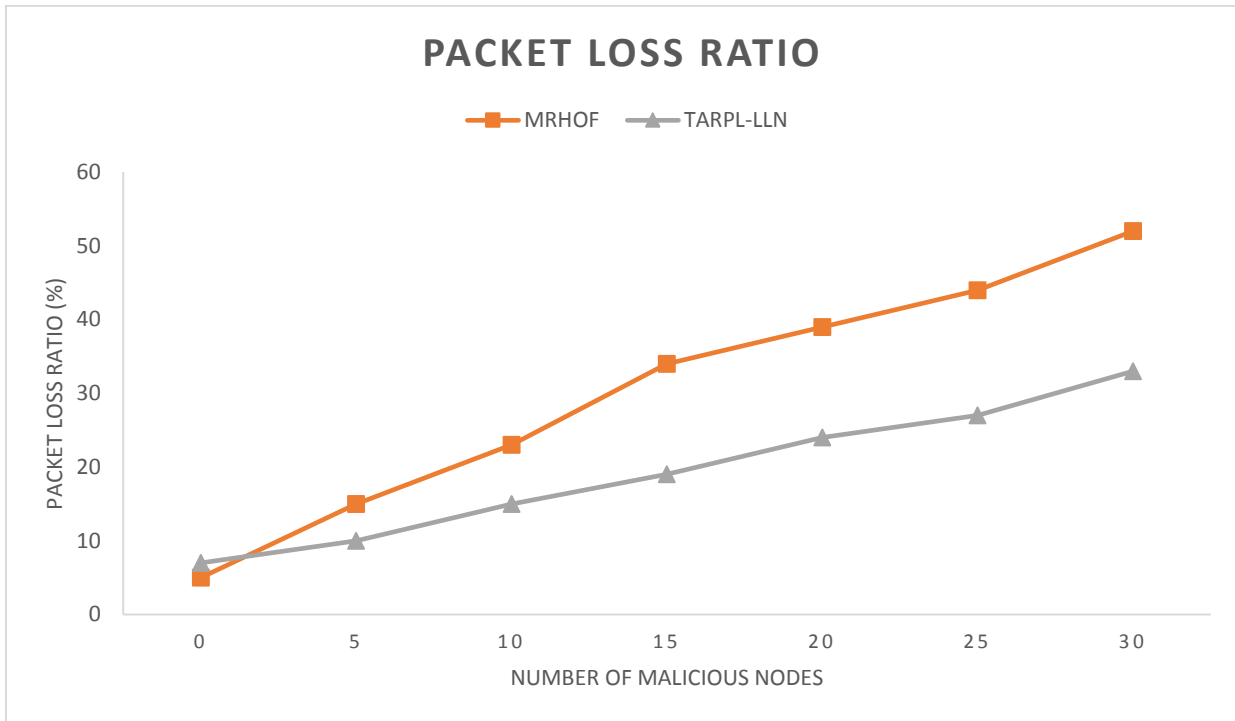


**Figure 5.11: Throughput of our approach and MRHOF objective function.**

We conclude from this graph that as the number of malicious nodes in the network starts to increase the throughput starts to decrease. Both curves decrease, but the one which represents MRHOF decreases critically, where it touches 1 bps when 30% of the nodes are malicious. However, in TARPL-LLN, the throughput is still acceptable whenever the number of malicious nodes increase.

### 5.5.5. Packet Loss Ratio

Here, we are testing the ratio of the lost packets, where it is calculated based on the number of packets sent by the nodes and the number of packets that did reach the sink node. Figure 5.12 illustrates the packet loss ratio in the network while varying the number of the malicious nodes.



**Figure 5.12: Packet loss ratio of our approach and MRHOF objective function.**

As we can notice, the graph above shows that in TARPL-LLN when the percentage of malicious nodes became 30, the packet loss ratio is still acceptable (33%) comparing to the MRHOF objective function where it is equal to 52%. Therefore, we can clearly say that our approach decreases dramatically the number of packets lost.

## 5.6. Comparing TARPL-LLN with 2 other approaches

We showed in the previous experiments that TARPL-LLN performs better than the standard routing protocol with the presence of malicious nodes in the network. In this subsection, we will do similar experiments to compare our approach with two of the previous approaches which are specified as related works for our thesis (“trust-based service management for social IoT system” (social) and “using trust management to defend against routing distribution attacks” (PFI)). We use Remaining Energy, packet latency and percentage of malicious detection as performance metrics for the comparison.

### 5.6.1. Remaining Energy

In this subsection, we study the average remaining energy in the network while varying the percentage of malicious nodes in the network consisted of 100 nodes. The percentage of malicious nodes varies from 0 to 30 percent of the network. Figure 5.13 illustrates the remaining energy in the network using TARPL-LLN, standard RPL, social approach and the approach that defends against routing distribution attacks (PFI).

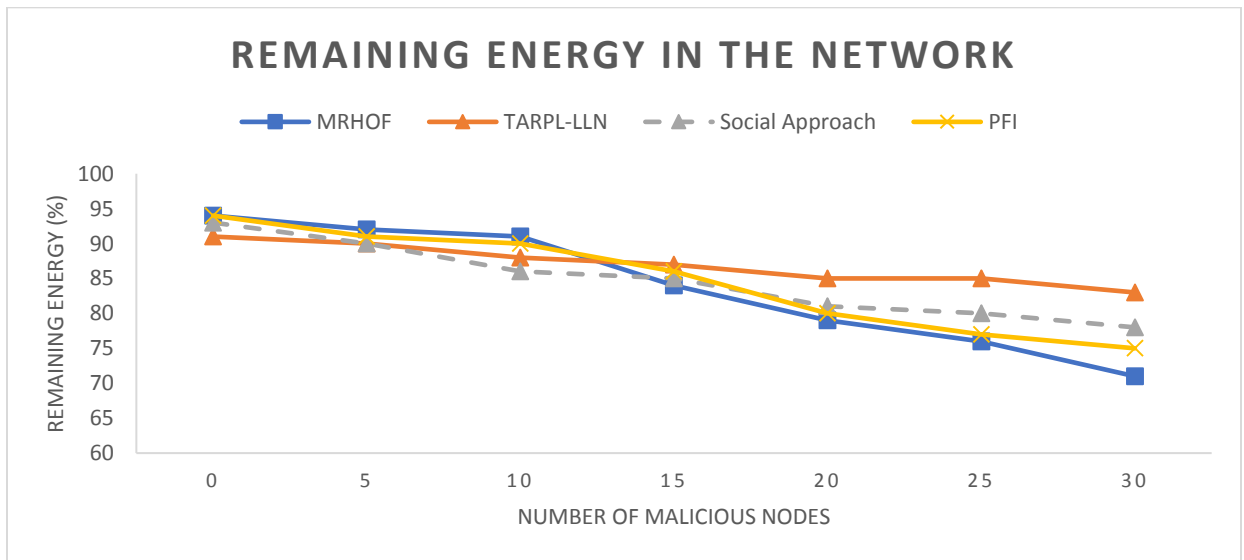
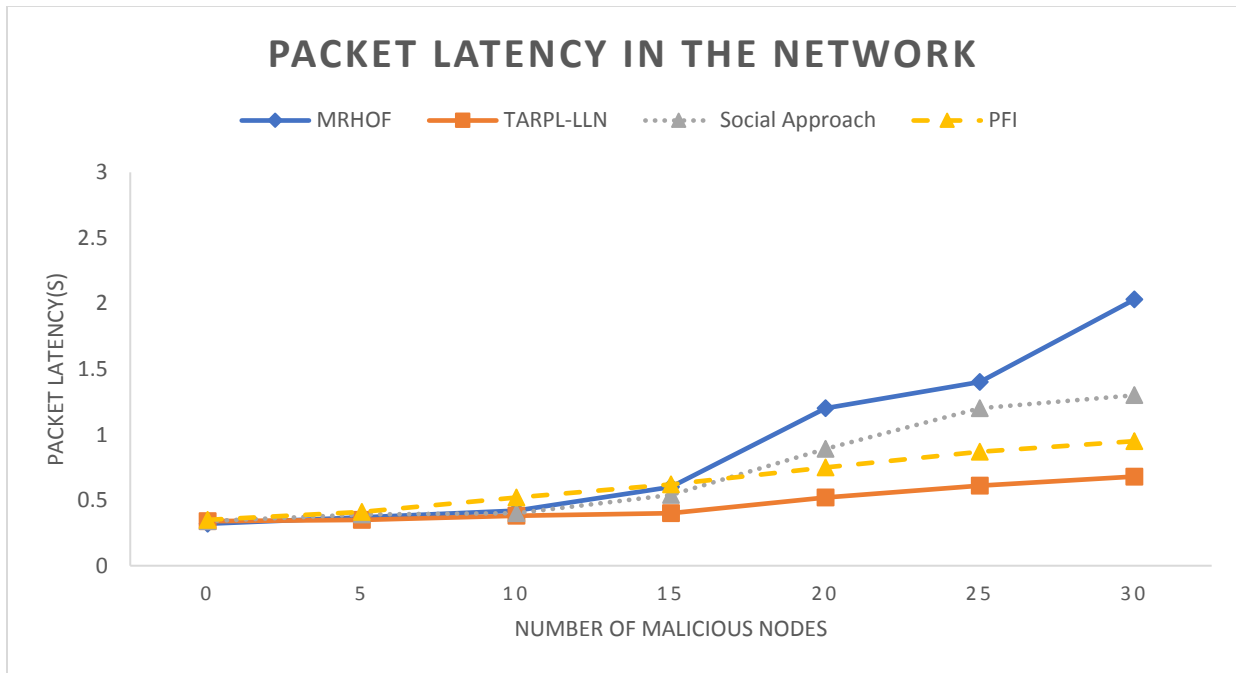


Figure 5.13: Remaining energy in the network while varying number of malicious nodes.

We notice that the remaining energy in the network that uses TAPRL-LLN has the highest remaining energy when the number of malicious nodes increase. This is because, in our approach, we are detecting more malicious nodes where this leads to less retransmission of lost packets. However, when the number of misbehaved nodes is small, we can see that all approaches and the standard routing protocol achieved almost the same performance.

### 5.6.2. Packet Latency

As in the previous subsection (5.5.3), we are studying here the packet latency in the network while varying the percentage of malicious nodes in the network to check which approach performs better in terms of packet latency.



**Figure 5.14: Packet Latency in the network while varying number of malicious nodes.**

In the figure 5.14, we notice that TARPL-LLN still has the lowest packet latency since it can detect more malicious nodes. In addition, the “using trust management to defend against

routing distribution attacks” approach performs well because it has the ability to detect all the distribution attacks like selective and black whole attacks.

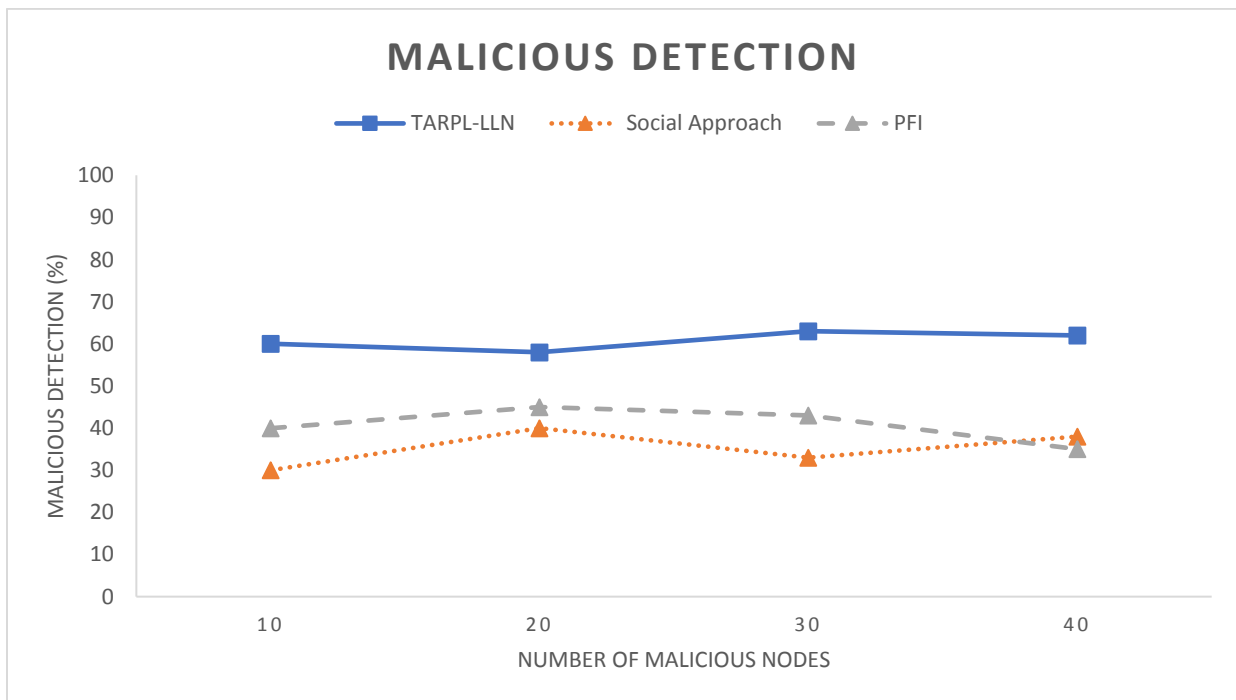
### 5.6.3. Percentage of Malicious Detection

In this simulation experiment, we study the malicious detection percentage. We define the malicious detection percentage as follows:

$$MaliciousDetection = \frac{T_{Detected}}{T_{Malicious}} * 100$$

where  $T_{Detected}$  is defined as the total number of malicious nodes detected and  $T_{Malicious}$  is defined as the total number of malicious nodes in the system.

Figure 5.15 shows the percentage of detected malicious nodes while varying the number of malicious nodes from 10 to 40 in a network consists of 100 nodes.



**Figure 5.15: percentage of malicious nodes detected while varying the number of malicious nodes. Network size= 100, request rate= 1 packet/ 2seconds, simulation time = 1 hour.**

In this graph, TARPL-LLN was able to detect more malicious nodes than other approaches. It was able to detect up to 60% of malicious nodes regardless of the number of malicious nodes in the network. The two other approaches were able to detect up to 40% but this percentage is not stable as ours with different percentage of malicious nodes.

Note that not all the malicious nodes act maliciously all the time or most of the time. This lowers the percentage of malicious detection.

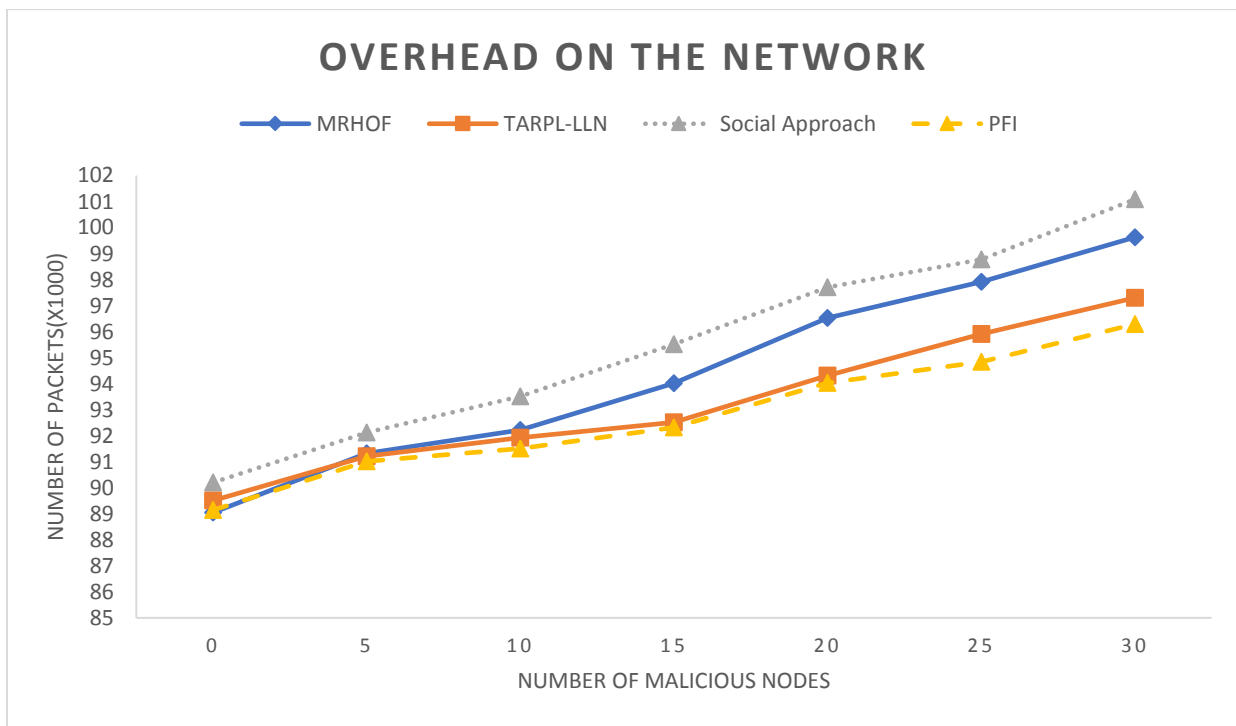
### **5.7. Overhead on The Network**

As noticed, TARPL-LLN increases the size of DIO message since each node updates the message by adding trust values of its neighbors. Hence more packets will be transmitted in the network, and consequently this could make overhead on IoT network. Therefore, the last experiment was to test whether TARPL-LLN can lead to an overhead on the IoT network.

In this experiment, we study the number of all transmitted packets in the network while varying the percentage of malicious nodes from 0 to 30 percent in a network consists of 100 nodes for the standard RPL, TARPL-LLN, Social approach and PFI. In addition, because the retransmissions of packets are due to the selective and black hole attacks, we include only these 2 types of attacks in the network.

Figure 5.16 shows that when there is no malicious node, standard RPL has the lowest number of packets in the network. This is because there is no addition metrics in the control messages and the packets are not retransmitted. Even though in our approach TARPL-LLN there are some new trust metrics, this doesn't impact much the number of packets transmitted, because the metrics will be stored only in the control messages wherein these messages are transmitted based on a trickle timer. As the number of malicious nodes increase, number of packets transmitted in TARPL-LLN and PFI becomes much less than those in social approach and standard RPL. This

is expected, since the social approach and standard RPL don't have capacity to detect this type of attack, also in social approach the DIO message contains the trust values for all nodes in the network. However, in PFI and TARPL-LLN, the number of packets transmitted is less, since they can detect selective and black hole attacks, thus a smaller number of packets will be retransmitted. Finally, we notice that the number of transmitted packets in the case of PFI is less than ours, where this is due to the usage of less number of metrics in the control messages where PFI can only detect this type of attacks.



**Figure 5.16: Number of packets transmitted in the network while varying the number of malicious nodes. Network size= 100, simulation time = 1 hour, types of attacks: selective and black hole.**

### 5.8. Effect of $\alpha$ and $\beta$ on trust Evaluation

The equations used in TARPL-LLN contain five different parameters which are used to weigh between the metrics. In equation 4.15, the node calculates the social trust values for its neighbors based on three different metrics which are *“honesty, intimacy and cooperativeness”*,



hence we give each metric a weight where the summation for these 3 weights ( $w_1, w_2$  and  $w_3$ ) is equal to 1. For our experiments, since we need to deal with all attacks equally, we give these 3  $w$ 's same value which is  $\frac{1}{3}$ . Regarding the two other parameters ( $\alpha$  and  $\beta$ ), in the section we will do new experiments to evaluate their effects on the calculation of the trust values.

### **5.8.1. Effect of $\alpha$ on trust evaluation**

We first explore the effect of parameter  $\alpha$  on social trust evaluation.  $\alpha$  represents the weight associated with direct trust with respect to past experience in equation 4.16. To do the experiments, we vary the value of  $\alpha$  by selecting different values (0.1, 0.3, 0.5, 0.8) and set the value of  $\beta$  to 0 in order to isolate its effect. Note that the percentage of malicious nodes is 30 percent, the time of the experiment is 6 hours and for the sake of the experiment we set the trust value for the nodes at time zero equals to 0.5.

Figure 5.17 shows the results of social trust evaluation toward a malicious node randomly selected whose status becomes malicious after 3 hours. We notice that after the status changes, the social trust evaluation starts to converge towards zero. As the value of  $\alpha$  increases, the trust evaluation converges in all cases to zero faster but with a higher fluctuation. Therefore, there is an inherent trade-off between trust convergence time versus trust fluctuation where as the value of  $\alpha$  increases, the trust value converges to zero faster, but the trust fluctuation also becomes higher.

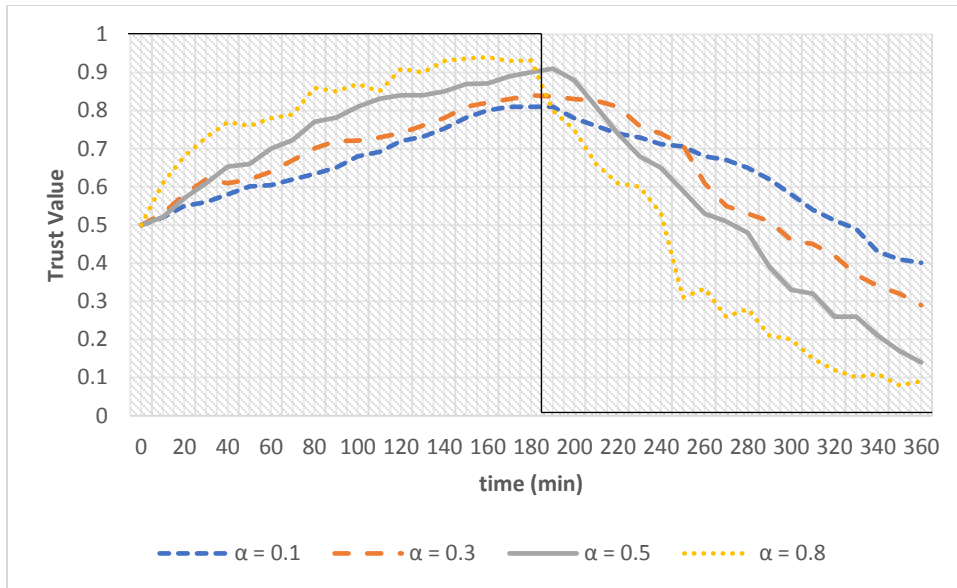


Figure 5.17: Trust value of a malicious node randomly chosen

### 5.8.2. Effect of $\beta$ on trust evaluation

In this subsection, we investigate the effect of parameter  $\beta$  on social trust value. Recall that  $\beta$  is the weight associated with indirect recommendation with respect to past experience in equation 4.17. In order to do the experiment, we vary the value of  $\beta$  by selecting different values  $\{0.1, 0.3, 0.5, 0.8\}$  and set the value of  $\alpha$  to 0.5 to isolate its effect.

As in the previous subsection, the percentage of malicious nodes is 30 percent, the time of the experiment is 6 hours and the malicious node is randomly selected whose status becomes malicious after 3 hours. Figure 5.18 shows that the social trust value converges quickly to 1 as the value of  $\beta$  increases. Initially using more recommendation (high value for  $\beta$ ) helps trust convergence quickly. However, after the status changes to malicious, we see that as the value of  $\beta$  is near to 0.5, the convergence toward zero is faster. This is because, the node will be calculating the trust values based on its observation and the other's recommendations.

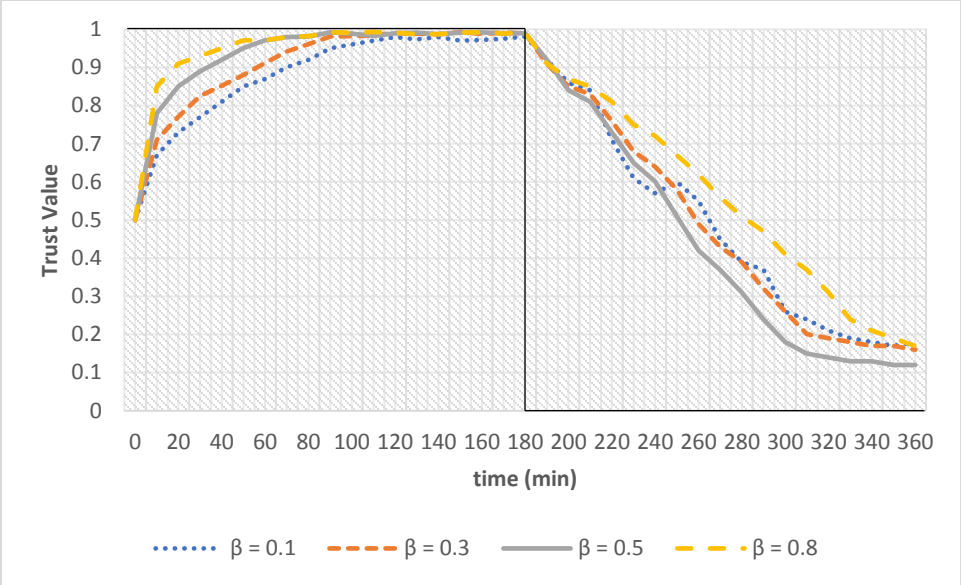


Figure 5.18: Trust value of a malicious node randomly chosen

## CHAPTER 6

### CONCLUSION

In this thesis, we have surveyed the routing protocol for low power and lossy network (RPL) that is used in internet of things (IoT). Then, we have examined its advantages and disadvantages showing that the security is a major issue in this routing protocol. To overcome this issue, we have proposed a trust aware routing protocol for low power and lossy network that is able to detect and isolate malicious nodes during the construction and maintenance of the network topology. The proposed approach is reputation-based where each node updates the trust values of its neighbors upon encounter or interaction events. The trust assessment is updated by both direct observations and indirect recommendations evaluated by the nodes' neighbors. In this approach, we divided the trust evaluation into two levels which are social and quality of service levels. The former is used to detect the malicious nodes that provide incorrect feedbacks about its neighbors wherein the latter is used to detect the nodes that don't behave correctly.

Our proposed approach was incorporated in RPL protocol, evaluated and compared with the standard protocol and two approaches found in the literature. During these simulations we have measured several parameters such as the energy consumption, packet delivery ratio, packet latency, throughput and percentage of malicious nodes detected. As a general remark, our approach proves to be efficient and well-organized in detecting several malicious nodes.

the limitation of the proposed approach is that a malicious node that performs Sybil and identity attack where it clones another node's identity cannot be detected. We assume that these attacks can be handled by authentication system that doesn't allow a node to mimic another node identity.

## REFERENCES:

- [1] Rolf H. Weber, and Romana Weber, "*Internet of things*". Vol. 12. Heidelberg: Springer, 2010, New York, USA.
- [2] Gope, Prosanta, and Tzonelih Hwang. "*BSN-Care: A secure IoT-based modern healthcare system using body sensor network*". In *IEEE Sensors Journal* 16.5 (2016): 1368-1376.
- [3] Li, Baoan, and Jianjun. "*Research and application on the smart home based on component technologies and Internet of Things*." In *Procedia Engineering journal*, 15 (2011): 2087-2092.
- [4] Vermesan, Ovidiu, et al. "*Internet of things strategic research roadmap*", In *Internet of Things-Global Technological and Societal Trends book*, 1.2011 (2011): 9-52.
- [5] Kushalnagar, Nandakishore, Gabriel Montenegro, and Christian Schumacher, "*IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals*". No. RFC, Internet Request for Comments, 4919. 2007.
- [6] Palattella, Maria Rita, et al. "*Standardized protocol stack for the internet of (important) things*". In *IEEE communications surveys & tutorials journal*, 15.3 (2013): 1389-1406.
- [7] Dhumane, Amol, Rajesh Prasad, and Jayashree Prasad. "*Routing issues in internet of things: a survey*." In *Proceedings of the International Multiconference of Engineers and Computer Scientists*, Hong Kong, China, pp. 16-18. 2016.
- [8] Xin, Hua-Mei, and Kun Yang. "*Routing protocols analysis for Internet of Things*." In *Proceedings of the 2nd International Conference on Information Science and Control Engineering*, Shanghai, China, April 2015, pp. 447-450.

- [9] Winter, Tim, et al. *RPL: "IPv6 routing protocol for low-power and lossy networks"*. No. RFC, Internet Request for Comments, 6550. 2012.
- [10] Thubert, Pascal. "*Objective function zero for the routing protocol for low-power and lossy networks (RPL)*". No. RFC, Internet Request for Comments, 6552. 2012.
- [11] Granjal, Jorge, Edmundo Monteiro, and Jorge Sá Silva. "*Security for the internet of things: a survey of existing protocols and open research issues.*" In IEEE Communications Surveys & Tutorials journal, 17.3 (2015): 1294-1312.
- [12] Pongle, Pavan, and Gurunath Chavan. "*A survey: Attacks on RPL and 6LoWPAN in IoT.*" In Proceedings the 2015 International Conference on Pervasive Computing (ICPC), Austin, Texas, USA, March 2015, pp. 1-6
- [13] Aris, Ahmet, Sema F. Oktug, and Berna Ors Yalcin. "*RPL version number attacks: In-depth study.*" In NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, IEEE, pp. 776-779.
- [14] Seeber, Sebastian, et al. "*Towards a trust computing architecture for RPL in Cyber Physical Systems*". In Proceedings of the 9th International Conference on Network and Service Management CNSM, Rio, Brazil, 2013, pp. 134-137
- [15] Chen, Ray, Fenye Bao, and Jia Guo. "*Trust-based service management for social internet of things systems*" In IEEE transactions on dependable and secure computing journal, 13(6), pp.684-696
- [16] Bao, Fenye, et al. "*Hierarchical trust management for wireless sensor networks and its application to trust-based routing.*" In Proceedings of the 2011 ACM Symposium on Applied Computing, TaiChung, Taiwan, March 21 - 24, 2011, pp. 1732-1738.

- [17] Djedjig, Nabil, Djamel Tandjaoui, and Faiza Medjek. "*Trust-based RPL for the Internet of Things*." In Proceedings of the twentieth IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, July 9, 2015, pp. 962-967.
- [18] Karkazis, Panagiotis, et al. "*Design of primary and composite routing metrics for rpl-compliant wireless sensor networks*". In Proceedings the 2012 International Conference on Telecommunications and Multimedia (TEMU), Heraklion, Greece, 25-27 July 2012, pp. 13-18.
- [19] Hou, Ling, et al. "*Using trust management to defend against routing disruption attacks for cognitive radio networks*", In Proceedings the 2016 IEEE International Conference on Consumer Electronics-China, Hongkong, China, pp. 1-4.
- [20] Lahbib, Asma, et al. "*Link reliable and trust aware RPL routing protocol for Internet of Things*." In Proceedings the 16th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 2017, pp. 1-5.
- [21] Felsche, Matthias, Alexander Huhn, and Horst Schwetlick. "*Routing protocols for 6LoWPAN*." In Proceedings of the 4th workshop on Embedded networked sensors, Cork, Ireland, June 25 – 26, 2011, pp. 78-82.
- [22] Kumar, Vinay, and Sudarshan Tiwari. "*Routing in IPv6 over low-power wireless personal area networks (6LoWPAN): A survey*." In Computer Networks and Communications 2012 journal, 2012.
- [23] W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks" ACM Comput. Surv., vol. 45, no. 4, p. 47, Aug. 2013.
- [24] Gnawali, Omprakash, and Philip Levis. *The minimum rank with hysteresis objective function*. No. RFC, Internet Request for Comments, 6719. 2012.

- [25] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks" In *2010 Proceedings IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–9.
- [26] Oran David, "*OSI IS-IS intra-domain routing protocol*". No. RFC, Internet Request for Comments, 1142. 1990
- [27] Moy and John T. "*OSPF: anatomy of an Internet routing protocol*" Addison-Wesley Professional, 1998.
- [28] Mallikarjun Swamy and Keshava Prasanna. "Markle Tree Based Authentication Protocol for Lifetime Enhancement in Wireless Sensor Networks." *Int. J. Advanced Networking and Applications*, 2017, 8(05), pp.3209-3212
- [29] Dvir, Amit, and Levente Buttyan. "*VeRA-version number and rank authentication in rpl*". In *Proceedings the Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, Vancouver, Canada, May 2011, pp. 709-714
- [30] Council, A. (2008). "*Global Trends 2025: A Transformed World*". Executive Summary of the National Intelligence Council report of the same name.
- [31] Stankovic, John A. "*Research directions for the internet of things*." *IEEE Internet of Things Journal* 1.1, 2014, pp. 3-9.
- [32] Vasseur, Jean-Philippe, et al. "*Routing metrics used for path calculation in low-power and lossy networks*". No. RFC, Internet Request for Comments, 6551. 2012.
- [33] Levis, P., Clausen, T., Hui, J., Gnawali, O., & Ko, J. "*The trickle algorithm*" No. RFC, Internet Request for Comments, 6206, 2011.
- [34] Djedjig, Nabil, et al. "*New trust metric for the rpl routing protocol*." In *Proceedings of the 8th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, Jan 26 2017, pp. 328-335.



- [35] Whiting, Doug, Russ Housley, and Niels Ferguson. "*Counter with cbc-mac (ccm)*". No. RFC, Internet Request for Comments, 3610. 2003.
- [36] Dunkels, Adam, Bjorn Gronvall, and Thiemo Voigt. "*Contiki-a lightweight and flexible operating system for tiny networked sensors.*" In Proceedings of the 29th annual IEEE international conference on local computer networks, Tampa, USA, Nov 16 2004, pp. 455-462.
- [37] Osterlind, Fredrik, et al. "*Cross-level sensor network simulation with cooja.*" In Proceedings of the first IEEE International Workshop on Practical Issues in Building Sensor Network Applications, 2006.
- [38] Dunkels, Adam, et al. "*Software-based on-line energy estimation for sensor nodes*". In Proceedings of the 4th workshop on Embedded networked sensors. Cork, Ireland, June 25 - 26, 2007
- [39] Dunkels, Adam, et al. "*Powertrace: Network-level power profiling for low-power wireless networks.*" (2011).
- [40] T.R. Andel and A. Yasinac. "*On the credibility of manet simulations*". IEEE Computer Society Press, Los Alamitos, CA, USA, vol. 39, 2006, pp. 48-54.
- [41] Haroon, Asma, et al. "*Constraints in the IoT: the world in 2020 and beyond.*" Constraints 7.11 (2016): 252-271.