

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université A.MIRA-BEJAIA



جامعة بجاية  
Tasdawit n Bgayet  
Université de Béjaïa

Faculté des Sciences Exactes  
Département Informatique

# THÈSE

Présentée par

**MEDJEK Faiza**

Pour l'obtention du grade de

**DOCTEUR EN SCIENCES**

Filière : Informatique

Option : Cloud Computing

Thème

**Intrusions detection and tolerance for the self-organising Internet of Things**

Soutenue le : 12/07/2022

Devant le Jury composé de :

<b>Nom et Prénom</b>	<b>Grade</b>		
<b>Mr TARI Abdelkamel</b>	Professeur	Univ. de Bejaia	Président
<b>Mr TANDJAOUI Djamel</b>	Directeur de recherche	CERIST, Alger	Rapporteur
<b>Mme MOUSSAOUI Samira</b>	Professeur	Univ. de USTHB	Examinatrice
<b>Mr GUERROUMI Mohamed</b>	MCA	Univ. de USTHB	Examinateur
<b>Mr FARAH Zoubeyr</b>	MCA	Univ. de Bejaia	Examinateur
<b>Mr ROMDHANI Imed</b>	Professeur Associé	Univ. de Napier	Invité

**Année Universitaire : 2021/2022**

# Acknowledgements

First and foremost, I thank Allah Almighty for giving me the strength, the courage and the knowledge to complete my doctoral degree.

I would like to express my deepest gratitude to my supervisor Research Director Djamel Tandjaoui, for his continuous support and guidance, monitoring, orientations and relevant remarks, which steered me in the right direction to complete my work.

I sincerely thank Dr Imed Romdhani for his support and valuable comments. It was a great pleasure and honour to work with him. He greeted me to his research laboratory at Napier University in Edinburgh, which helped me a lot to build a robust and useful research collaboration.

I would like to thank Prof. Tari Abdelkamel (University of Bejaia) for agreeing to chair my thesis jury, as well as Prof. Moussaoui Samira (USTHB), Dr. Guerroumi Mohamed (USTHB) and Dr. Farah Zoubeyr (University of Bejaia) for honouring me with their acceptance as examiners in my thesis jury.

I would like to express my deepest gratitude to my parents for their support, love, prayers, patience, and sacrifices. It is to them that I owe all the success in my life. I would also like to extend my sincere thanks to my sisters, brother, and my family in-laws especially my mother in-laws for their support and encouragements. Big ups for my lovely daughter Maria. Finally, I accomplish this recognition by thanking my lovely husband for giving me more than enough love, understanding, and strength necessary to complete my work.

I would like to warmly thank my friends especially Sabira OUADI for their appreciable help, their continuous encouragement and their uninterrupted moral support.

I would like to thank my colleagues from CERIST and Edinburgh Napier University for having encouraged, motivated and above all supported and helped me

before and during the preparation of my thesis.

And to be sure not to forget anyone, that all those, near or far, contributed by their advice, their encouragement or their friendship, to the result of this modest work, find here the expression of my deep recognition.

# Contents

<b>List of Abbreviations</b>	<b>xi</b>
<b>List of Publications</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Motivation and Problem Statement . . . . .	3
1.3 Contributions . . . . .	3
1.4 Organisation of the Thesis . . . . .	5
<b>2 Background: The Internet of Things, Intrusion Detection Systems, and Machine Learning</b>	<b>8</b>
2.1 The Internet of Things (IoT) . . . . .	8
2.1.1 IoT Definition . . . . .	8
2.1.2 IoT's Applications . . . . .	11
2.1.3 LLNs-IoT Characteristics . . . . .	11
2.2 Internet of Things Protocols . . . . .	13
2.2.1 IEEE 802.15.4 . . . . .	13
2.2.2 IPv6 Over LoW Power Wireless Area Networks (6LoWPAN) . . . . .	14
2.2.3 Routing Protocol for Low-Power and Lossy Networks (RPL) . . . . .	15
2.2.4 Constrained Application Protocol (CoAP) . . . . .	16
2.3 Security Challenges in IoT . . . . .	17
2.3.1 Perception/Sensing Layer . . . . .	18
2.3.2 Network Layer . . . . .	19
2.3.3 Application/Service Layer . . . . .	22
2.4 Intrusion Detection Systems (IDSs) . . . . .	22
2.4.1 IDSs Types . . . . .	23
2.4.2 IDSs Locations and Methods . . . . .	23
2.5 Machine Learning Concept . . . . .	25
2.5.1 ML Common Terminology . . . . .	25
2.5.2 ML Tasks . . . . .	26

2.5.3	ML Algorithms . . . . .	26
2.6	Summary . . . . .	31
<b>3</b>	<b>The IPv6 Routing Protocol for LLNs (RPL): Overview, Security Issues, and State-of-the-art Solutions</b>	<b>32</b>
3.1	RPL Topology Construction . . . . .	32
3.1.1	RPL Control Messages . . . . .	34
3.1.2	Upward routes . . . . .	35
3.1.3	Downward routes . . . . .	35
3.2	Objective Functions . . . . .	37
3.2.1	The Objective Function Zero (OF0) . . . . .	37
3.2.2	Minimum Rank with Hysteresis Objective Function (MRHOF)	38
3.2.3	Routing Metrics for Path Calculation . . . . .	38
3.3	Trickle Timer . . . . .	39
3.4	RPL Tools . . . . .	40
3.4.1	Operating Systems . . . . .	40
3.4.2	Cooja Simulator . . . . .	41
3.4.3	Hardware . . . . .	42
3.5	RPL Security . . . . .	42
3.5.1	Self-healing and Fault Tolerance Mechanisms . . . . .	42
3.5.2	Security Features . . . . .	43
3.6	RPL Vulnerabilities . . . . .	44
3.6.1	RPL Security Limitations . . . . .	44
3.6.2	Attacks against RPL . . . . .	45
3.7	Security Enhancements for RPL . . . . .	52
3.7.1	Cryptography-based Solutions . . . . .	52
3.7.2	Trust-based Solutions . . . . .	52
3.7.3	IDS-based Solutions . . . . .	53
3.8	Comparison and Discussion . . . . .	59
3.9	Summary . . . . .	65
<b>4</b>	<b>RPL's Performance under DIS and Sybil Attacks and New Approach for the Intrusions Tolerance</b>	<b>66</b>
4.1	The Multicast DIS Attack (M-DIS): Reminder . . . . .	66
4.2	The Sybil Attacks . . . . .	67
4.2.1	Sybil Attacks Overview . . . . .	67
4.2.2	RPL under Sybil Attacks . . . . .	68
4.3	Analytical-based RPL's Performance Evaluation under Sybil Attack	70
4.3.1	Control messages overhead . . . . .	71
4.3.2	Packets delivery . . . . .	75

4.3.3	Energy consumption . . . . .	75
4.3.4	Discussions . . . . .	76
4.4	Simulation-based RPL's Performance Evaluation under SybM Attack	77
4.4.1	Simulation Settings . . . . .	78
4.4.2	Simulations Results . . . . .	79
4.5	The Proposed Approach: RPL-MRC . . . . .	82
4.5.1	Response Delay . . . . .	83
4.5.2	Timer Readjustment . . . . .	84
4.6	Approach Evaluation . . . . .	85
4.6.1	Performance Metrics . . . . .	85
4.6.2	Simulation Settings . . . . .	86
4.6.3	The M-DIS Attack Frequency Effect . . . . .	87
4.6.4	The Number of Attackers Effect . . . . .	90
4.6.5	The MRC Parameter Effect . . . . .	91
4.6.6	The Data Packet Rate Effect . . . . .	94
4.7	Approach Evaluation under Mobility: SybM Case . . . . .	95
4.7.1	Control Overhead . . . . .	96
4.7.2	Power Consumption . . . . .	97
4.7.3	Data Packets Overhead . . . . .	97
4.8	Summary . . . . .	98
<b>5</b>	<b>Intrusion Detection and Tolerance Systems for RPL's Security</b>	<b>99</b>
5.1	A Trust-based Intrusion Detection System for Mobile RPL Based Networks . . . . .	99
5.1.1	T-IDS Characteristics . . . . .	100
5.1.2	T-IDS Actors . . . . .	101
5.1.3	T-IDS Modules . . . . .	103
5.1.4	T-IDS Advantages and Limitations . . . . .	109
5.2	Fault-Tolerant AI-Driven Intrusion Detection System for the Internet of Things . . . . .	109
5.2.1	RPL Intrusions . . . . .	110
5.2.2	ML Methods . . . . .	110
5.2.3	Performance Evaluation Metrics . . . . .	111
5.2.4	Dataset Generation . . . . .	112
5.2.5	Classifiers Evaluation and Discussion . . . . .	119
5.2.6	RF-Based Intrusion Detection System for RPL (RF-IDSR) . .	122
5.3	Summary . . . . .	132

<b>6</b>	<b>Conclusions and Perspectives</b>	<b>133</b>
6.1	Thesis Summary and Contributions Review . . . . .	133
6.2	Limitations and Future directions . . . . .	135

# List of Figures

2.1	IoT five-layer architecture [1]. . . . .	9
2.2	IoT three-layer architecture. . . . .	10
2.3	IoT application domains. . . . .	12
2.4	6LoWPAN architecture [2]. . . . .	15
2.5	6LoWPAN protocol stack. . . . .	15
2.6	CoAP functioning within an IoT environment [3]. . . . .	17
2.7	IoT security challenges according to [4]. . . . .	18
2.8	Advantages and disadvantages of anomaly-based detection techniques [1]. . . . .	25
2.9	Decision tree technique illustration. . . . .	27
2.10	Random forests technique. . . . .	28
2.11	K-nearest neighbour (KNN) technique. . . . .	28
2.12	Naïve Bayes technique. . . . .	29
2.13	MLP technique. . . . .	30
2.14	Logistic regression technique. . . . .	30
2.15	Shallow vs DL. . . . .	31
3.1	The routing protocol for low power and lossy networks. . . . .	33
3.2	DIO format. . . . .	34
3.3	DAO format. . . . .	34
3.4	DIS format. . . . .	35
3.5	RPL DODAG and upward routes construction. . . . .	36
3.6	Routing in RPL. Existing routes are shown next to the network nodes [5]. . . . .	37
3.7	The Trickle algorithm for a node [6]. . . . .	40
3.8	ContikiOS's core and loaded programs [7]. . . . .	41
3.9	Specifications of typical constrained devices [8]. . . . .	42
4.1	Multicast DIS attack illustration. . . . .	67
4.2	SyBM model, where 6 attackers move periodically across their neigh- bours towards the BR while multicast DIS messages. . . . .	69



4.3	$N_{DIOsent}$ and $N_{DIOreceived}$ within the network. . . . .	73
4.4	DIS/DAO messages overhead. . . . .	74
4.5	DIO messages overhead. . . . .	74
4.6	Control messages overhead. . . . .	74
4.7	Sybm's energy cost. . . . .	76
4.8	Sybm's energy cost for $K=1$ , $K=3$ and $K=5$ . . . . .	77
4.9	The experimental network topology under Sybm attack in the case of 10 malicious nodes. . . . .	80
4.10	Sybm attacks control overhead . . . . .	80
4.11	Control overhead vs number of attacker . . . . .	81
4.12	Energy cost vs number of attacker . . . . .	82
4.13	Packet delivery ratio vs number of attacker . . . . .	83
4.14	New DIO Message . . . . .	84
4.15	The Trickle Timer on Receiving Multicast DIS Message . . . . .	85
4.16	Control overhead vs attack frequency. . . . .	88
4.17	Power consumption vs attack frequency. . . . .	89
4.18	Delivered and duplicate data packets vs attack frequency. . . . .	89
4.19	Control overhead vs number of attackers. . . . .	90
4.20	Power consumption vs number of attackers. . . . .	91
4.21	Delivered and duplicate data packets vs number of attackers. . . . .	92
4.22	Number of DIO under different MRC values. . . . .	92
4.23	Power consumption under different MRC values. . . . .	93
4.24	Delivered and duplicate data packets for different MRC values. . . . .	94
4.25	Number of DIO vs data packet rate. . . . .	94
4.26	Power consumption vs data packet rate. . . . .	95
4.27	Delivered, duplicate and lost data packets vs data packet rate. . . . .	96
4.28	Control overhead under Sybm attack. . . . .	97
4.29	Power consumption under Sybm attack. . . . .	97
4.30	Data packets overhead under Sybm attack. . . . .	98
5.1	T-IDS architecture. . . . .	100
5.2	New DIO message format. . . . .	101
5.3	The backbone station operations. . . . .	102
5.4	The border router operations. . . . .	103
5.5	In-network nodes operations. . . . .	104
5.6	Feature selection process. . . . .	117
5.7	Feature importance datagram for SF dataset. . . . .	118
5.8	Correlations between different features for the SF dataset. . . . .	118
5.9	Classifiers performance per dataset for 60 minutes simulation time. . . . .	120

5.10	Fitting time for BH dataset vs VN dataset for 10mn simulation time.	121
5.11	MLP performance for VN datasets: 10mn vs 60mn simulation time. .	122
5.12	DL-model for VN dataset. . . . .	123
5.13	7-class classification performance. . . . .	123
5.14	RF-IDSR architecture. . . . .	125
5.15	Packet delivery ratio vs scenarios. . . . .	129
5.16	Control overhead vs scenarios. . . . .	129
5.17	Control overhead for the neighbours of the intruder 17. . . . .	130
5.18	Average power consumption vs scenarios. . . . .	131
5.19	Average power consumption for the neighbours of the intruder 17. .	131

# List of Tables

3.1	IDS solutions for RPL Networks 2011-2020 ... (Part 1)	63
3.2	IDS solutions for RPL Networks 2011-2020 ... (Part 2)	64
4.1	Terminology	71
4.2	Energy consumption on constrained node [9].	75
4.3	Energy cost for SybM (K=5 and M=10).	75
4.4	Energy cost for SybM (K=1 and M=10) corresponding to the energy cost of DIS attack.	76
4.5	Energy cost (J) for SybM attack depending on numbers of Sybil nodes and attackers.	76
4.6	Simulation parameters for SybM attack's effects on RPL	78
4.7	Scenarios	79
4.8	Simulation Parameters	88
4.9	Simulation Parameters for RPL-MRC under SybM Attack	96
5.1	Confusion matrix	111
5.2	Simulation Parameters	113
5.3	Features to be used for RPL's intrusions detection	115
5.4	The generated datasets for the IDS use	116
5.5	Features per dataset after data pre-processing	119
5.6	Simulation parameters for VN attack tolerance and detection	128

# List of abbreviations

**6LoWPAN** IPv6 over Low-Power Wireless Personal Area Networks

**AODV** Ad hoc On-Demand Distance Vector

**BLE** Bluetooth Low Energy

**BR** Border Router

**CM** Confusion Matrix

**CoAP** Constrained Application Protocol

**CSMA-CA** Carrier Sense Multiple Access with Collision Avoidance

**DAG** Directed Acyclic Graph

**DAO** Destination Advertisement Object

**DAO-ACK** Destination Advertisement Object Acknowledgement

**DDoS** Distributed Denial of Service

**DIO** DODAG Information Object

**DIS** DODAG Information Solicitation

**DL** Deep Learning

**DODAG** Destination Oriented Directed Acyclic Graph

**DoS** Denial of Service

**DT** Decision Tree

**ETX** Expected Transmission Count

**FFD** Full-Function Device

**Hydro** Hybrid Routing Protocol for LLNs

**ICMPv6** Internet Control Message Protocol for IPv6

**IDS** Intrusion Detection System

**IEEE** Institute of Electrical and Electronics Engineers

**IETF** Internet Engineering Task Force

**IoT** Internet of Things

**IPv6** Internet Protocol version 6

**KNN** K-Nearest Neighbour

**LLNs** Low-power and Lossy Networks

**LoWPAN** Low Power Wireless Personal Area Network

**LR** Logistic Regression

**LSD** Lightweight Sybil Attack Detection Framework

**M2M** Machine-to-Machine

**M-DIS** Multicast DODAG Information Solicitation

**MAC** Media Access Control

**MANET** Mobile Ad Hoc Network

**MRC** Maximum Response Code

**MRD** Maximum Response Delay

**ML** Machine Learning

**MLP** Multi-Layer Perceptron

**MP2P** MultiPoint-to-Point

**MRHOF** Minimum Rank with Hysteresis Objective Function

**MTU** Maximum Transmission Unit

**NB** Naive Bayes

**NFC** Near Field Communications

**OF** Objective Function

**OF0** Objective Function Zero

**PAN** Personal Area Network

**PDR** Packet Delivery Ratio

**PP** Preferred Parent

**P2P** Point-to-Point

**P2MP** Point-to-MultiPoint

**QoS** Quality of Service

**RF** Random Forests

**RF-IDS** Random Forests based IDS for RPL

**RFD** Reduced-Function Device

**RFID** Radio Frequency Identification

**ROLL** Routing Over Low-power and Lossy networks

**RPL** IPv6 Routing Protocol for Low-power and Lossy networks

**RPL-MRC** RPL Maximum Response Code

**RSSI** Received Signal Strength Indicator

**SIoT** Social Internet of Things

**SybilM** Sybil Mobile attack

**TDMA** Time Domain Multiple Access

**UDGM** Unit Disk Graph Medium

**UDP** User Datagram Protocol

**WSN** wireless sensor networks

# List of Publications

The following is a list of publications made while working on this thesis.

## First author

### Peer-reviewed Journal Publications

- **F. Medjek**, D. Tandjaoui, N. Djedjig, I. Romdhani, Fault-tolerant ai-driven intrusion detection system for the internet of things, *International Journal of Critical Infrastructure Protection (IJCIP)* 34 (2021) 100436.
- **F. Medjek**, D. Tandjaoui, N. Djedjig, I. Romdhani, Multicast DIS Attack Mitigation in RPL-Based IoT-LLNs, *Journal of Information Security and Applications (JISAS)* 61 (2021) 102939.

### Peer-reviewed Conference, and Workshop Publications

- **F. Medjek**, D. Tandjaoui, M. R. Abdmeziem, and N. Djedjig, Analytical evaluation of the impacts of sybil attacks against rpl under mobility, in: 2015 IEEE 12th International Symposium on Programming and Systems (ISPS), 2015, pp. 1-9.
- **F. Medjek**, D. Tandjaoui, I. Romdhani, N. Djedjig, Performance evaluation of rpl protocol under mobile sybil attacks, in: 2017 IEEE Trustcom/BigDataSE/ICSS, 2017, pp. 1049-1055. doi:10.1109/Trustcom/BigDataSE/ICSS.2017.351
- **F. Medjek**, D. Tandjaoui, I. Romdhani, and N. Djedjig, A trust-based intrusion detection system for mobile rpl based networks, in: 2017 IEEE 10th International Conference on Internet of Things (iThings-2017), 2017.

### Peer-reviewed Book Chapter Publications

- **F. Medjek**, D. Tandjaoui, I. Romdhani, N. Djedjig, Security threats in the internet of things: Rpl's attacks and countermeasures, in: *Security and Privacy*

in Smart Sensor Networks, IGI Global, 2018, pp. 147-178. doi:10.4018/978-1-5225-5736-4.ch008.

## Co-author

### Peer-reviewed Journal Publications

- N. Djedjig, D. Tandjaoui, **F. Medjek**, I. Romdhani, Trust-aware and cooperative routing protocol for iot security, Journal of Information Security and Applications 52 (2020) 102467.

### Peer-reviewed Conference, and Workshop Publications

- N. Djedjig, D. Tandjaoui, **F. Medjek**, Trust-based rpl for the internet of things, in: 2015 IEEE Symposium on Computers and Communication (ISCC), IEEE, 2015, pp. 962-967.
- N. Djedjig, D. Tandjaoui, I. Romdhani, **F. Medjek**, Trust-based defence model against mac unfairness attacks for iot, ICWMC 2017 127 (2017).
- N. Djedjig, D. Tandjaoui, **F. Medjek**, I. Romdhani, New trust metric for the rpl routing protocol, in: Information and Communication Systems (ICICS), 2017 8th International Conference on, IEEE, 2017, pp. 328-335.

### Peer-reviewed Book Chapter Publications

- N. Djedjig, D. Tandjaoui, I. Romdhani, **F. Medjek**, Trust Management in the Internet of Things, in: Security and Privacy in Smart Sensor Networks, IGI Global, 2018, pp. 122-146.



# Chapter 1

## Introduction

### 1.1 Background

The Internet of Things concept has been established since the founding of the Auto-ID Center at the Massachusetts Institute of Technology (MIT) in 1999. The Auto-ID Center created the electronic product code (EPC) number, which depends on radio frequency identification (RFID), in 2003. This idea is the crucial technology of the Internet of Things (IoT)[10]. In the IoT concept, everything real becomes virtual, which means that every person and physical object has a locatable, addressable, and readable counterpart on the Internet. These virtual entities are interconnected with each other exploiting their standard underlying technologies, such as ubiquitous and pervasive computing, embedded devices, communication technologies, sensor networks, and Internet protocols [11][12]. Besides, the entities in IoT can produce and consume services and collaborate toward a common goal thus allowing providing various applications.

Actually, IoT is used in almost all fields such as assisted living, e-health, automation and industrial manufacturing, logistics, intelligent transportation of people and goods, prediction of natural disasters, agriculture application, etc. It has been predicted that there will be billions of IoT smart objects connected to the Internet generating more than 45% of the entire Internet traffic [11][3]. The report by Forrester <sup>1</sup> states that in 2010 the number of devices connected to the Internet surpassed the earth's human population. It predicts that there will be a huge growth in the IoT industry in the next years. Furthermore, the report by Symantec <sup>2</sup> states that there will be up to 21 billion connected devices by 2020 where cities and companies will start adopting smart technologies in their operations. Garner <sup>3</sup> reports that more than 26.66 billion IoT devices were active in 2020, and it is expected that there will

---

<sup>1</sup><https://www.forrester.com>

<sup>2</sup><https://us.norton.com/>

<sup>3</sup><https://www.gartner.com>

be 75 billion IoT devices in the world by 2025.

Noticeably, the numerous noise presences across almost all industrial scenarios make the use of Low-Power and Lossy Networks (LLNs) inevitable for IoT. Indeed, LLNs are one of the main building blocks of the IoT [2]. They are made of a collection of interconnected embedded resource-constrained devices, such as sensor nodes with low computational and storage capabilities and are often battery operated. In addition, communication technologies are subject to high packet loss, frame size limitations, low data rates, short communication ranges, and dynamically changing network topologies. Another challenge faced by LLNs is their isolation from the IP world, as IP was not developed to be used considering the LLNs' limitations. To address this issues, the IPv6 Low Power Wireless Personal Area Network (LoWPAN) Working Group (WG) has been created by the Internet protocols standardisation body IETF (Internet Engineering Task Force) to introduce the IPv6 over Low-power Wireless Personal Area Network (6LoWPAN) standard, thus allowing the support of IP-based networking over constrained communication technologies such as the IEEE 802.15.4 standard [13][14][15][16][17].

Given that routing protocol is one of the main pillars of networking architecture, and because the development of efficient routing solutions for LLNs is crucial [16] [18] [17], several attempts have been proposed like the Collection Tree Protocol (CTP) [19] and the Hybrid Routing Protocol for LLNs (Hydro) [20]. It has been found that these protocols are too inefficient for satisfying requirements of overhead, power, reliability and latency [18]. Ultimately, the IETF Routing Over LLNs (ROLL) WG has designed and standardised the Routing Protocol for LLNs, namely the Routing Protocol for Low-Power and Lossy Networks (RPL) [21] [22]. The ROLL WG focuses only on routing for general IPv6 and 6LoWPAN networks. Thus, the terminology used in ROLL differs from 6LoWPAN terminology, for example LLN instead of LoWPAN and (LLN) Border Router instead of (LoWPAN) Edge Router.

RPL is a distance vector routing protocol that organises the physical network into a logical representation. It builds and maintains a Destination Oriented Directed Acyclic Graph (DODAG) topology using new ICMPv6 control messages. Besides, RPL provides self-organising and self-healing mechanisms. Nonetheless, RPL is characterised by its resource-constrained nature and the lack of tamper resistance. Furthermore, it does not consider nor ensure the network security through routing repair mechanisms, and the network will be unable to respond in a timely manner after an attack. Consequently, malicious nodes can exploit the RPL's operations to trigger several attacks that can damage severely the LLNs.

## 1.2 Motivation and Problem Statement

LLNs is a key actor in the realisation of the IoT concept allowing the deployment of a wide spectrum of services and applications that would make the human life easier and more flexible. However, though IoT inherits all the advantages of LLNs and sensor networks, unfortunately, it inherits the disadvantages being vulnerable to external attacks from the Internet and internal attacks from the LLNs [11]. Indeed, compromising a single object and/or communication channel in an LLN can paralyse the part or complete network. Specifically, a compromised device may exploit RPL control messages and operations to cause other devices perform heavy computations and disrupt the established network routes. Clearly, the increasing deployment of IoT in all emerging sectors and the danger exposed by these connected things render enabling reliable, secure, and intrusion-tolerant routing inevitable in such unpredictable environment [11][23]. Actually, security and manageability are extremely important as LLNs are typically autonomous.

Since the introduction of RPL, several studies have reported that it suffers from various limitations and weaknesses that make it vulnerable to a large spectrum of attacks (i.e., intrusions) [24][25][26][27]. The traditional mechanisms such as cryptography can be used to protect RPL from outside attackers. However, they are insufficient and inadequate to protect RPL from inside attackers. As a consequence, intrusion detection and tolerance mechanisms are required to secure RPL.

Indeed, objects should be able to defend themselves against network attacks. As a result, routing protocols such as RPL should incorporate mechanisms that respond to abnormal situations and allow objects to be able to use intrusion detection systems (IDSs) and other defensive mechanisms to ward off attacks. Thus, IDSs should be used as the last line of defence. Besides, IDSs can be adaptable depending on needs and can be enhanced with Machine Learning (ML) techniques in addition to other advanced technologies.

Therefore, the aim of this thesis is to address the security gaps of the RPL standard as it is one of the most popular routing protocol for resource constrained networks in the context of IoT. The main objective is to enhance RPL with intrusions detection and tolerance mechanisms that are based on ML and RPL specification to immediately detect and/or respond to potential threats.

## 1.3 Contributions

Considering the need for a standardised and secure IoT ecosystem especially with the significant growth in the number of connected devices, RPL has emerged as a key protocol to maintain routing for IoT-LLNs. Nevertheless, the latter has many

---

security issues that need to be tackled. Therefore, we focus our contributions on the RPL standard security enhancement. We began by conducting a comprehensive review of the IoT and RPL security issues that results in the three following contributions:

- **F. Medjek**, D. Tandjaoui, M. R. Abdmeziem, and N. Djedjig, Analytical evaluation of the impacts of sybil attacks against rpl under mobility, in: 2015 IEEE 12th International Symposium on Programming and Systems (ISPS), 2015, pp. 1–9. [28] .
- **F. Medjek**, D. Tandjaoui, I. Romdhani, N. Djedjig, Performance evaluation of rpl protocol under mobile sybil attacks, in: 2017 IEEE Trustcom/BigDataSE/ICISS, 2017, pp. 1049–1055. doi:10.1109/Trustcom/BigDataSE/ICISS.2017.351. [29].
- **F. Medjek**, D. Tandjaoui, I. Romdhani, N. Djedjig, Security threats in the internet of things: Rpl’s attacks and countermeasures, in: Security and Privacy in Smart Sensor Networks, IGI Global, 2018, pp. 147–178. doi:10.4018/978-1-5225-5736-4.ch008. [27].

In the first and second papers we introduced a new attack against RPL, named SybM attack, and evaluated its impacts on the RPL-based networks analytically [28] and with simulation [29]. The third paper is a book chapter that provides necessary details to understand IoT and its security issues, while focussing on RPL security. It starts by presenting the IoT applications, characteristics, standardised protocols, and security requirements. Then, it lists the attacks corresponding to each layer of the IoT three-layer architecture with a particular focus on the network layer and the routing protocol. Furthermore, the paper surveys, analyses, and classifies the existing and new threats against RPL. It discusses countermeasures and IDS solutions to tackle RPL attacks.

We proposed a first IDS approach for RPL security based on trust that has been the subject of the following publication:

- **F. Medjek**, D. Tandjaoui, I. Romdhani, and N. Djedjig, A trust-based intrusion detection system for mobile rpl based networks, in: 2017 IEEE 10th International Conference on Internet of Things (iThings-2017), 2017 [30].

In this work, we proposed an architecture for a trust-based IDS for RPL under mobility constraint, named T-IDS. The key focus of the IDS is to detect and counter the SybM attack defined in our previous works. We introduced the actors in T-IDS and detailed each module of our IDS strategy. First, we described the identity and mobility management modules. Then, we presented the IDS module that is enriched with three appending: i) a trust-based RPL scheme to select only trusted nodes for

routing, ii) a multicast defence scheme to prevent and tolerate multicast-related attacks such as SybM, and iii) a cross-layer scheme to eliminate malicious nodes at the link layer.

As T-IDS is a specification-based IDS, it can only detect specific attacks. Nevertheless, there are continues novelties in terms of threats against RPL, which necessitates a more sophisticated IDS such as the anomaly-based one. The next step was the following journal paper:

- **F. Medjek**, D. Tandjaoui, N. Djedjig, I. Romdhani, Fault-tolerant ai-driven intrusion detection system for the internet of things, *International Journal of Critical Infrastructure Protection (IJCIP)* 34 (2021) 100436.

In this work, we defined the RPL attacks that threaten the most its functionalities. In order to implement a security mechanism based on ML, we needed a dataset to use for the classifiers training. However, there is a lack of availability or privacy of developed RPL related datasets. Consequently, we implemented different attacks and generated one-class and multi-class datasets for RPL. In order to choose the best ML algorithm for our needs, we experimented various algorithms in a binary and multiple classification using the generated dataset. In addition, we designed RF-IDSR, the intrusions detection and tolerance system for RPL. RF-IDSR has two parts; intrusion detection part and intrusion tolerance part. On the one hand, the intrusion detection part implements the ML-based IDS to detect several routing attacks. On the other hand, the intrusion tolerance part implements mechanisms (algorithms) to tolerate specific routing attacks against RPL. The latter allow RPL itself to reduce or eliminate the effects of the attacks on the network's performance.

As continuity to our work, we introduced and implemented an intrusion tolerance mechanism named RPL-MRC to mitigate multicast-related attacks. Our last contribution was the following:

- **F. Medjek**, D. Tandjaoui, N. Djedjig, I. Romdhani, Multicast DIS Attack Mitigation in RPL-Based IoT-LLNs, *Journal of Information Security and Applications (JISAS)* 61 (2021) 102939.

The proposed approach introduces a response delay mechanism and an RPL's timer readjustment mechanism to tolerate and counter Multicast related attacks.

## 1.4 Organisation of the Thesis

The remainder of this thesis is organised into four chapters as follows:

---

## **Chapter 2: Background: The Internet of Things, Intrusion Detection Systems, and Machine Learning**

Chapter 2 presents a background on the IoT paradigm; definitions and applications, characteristics and standardised protocols, and its security challenges. In addition, this chapter provides the preliminary information about the definitions relevant to IDSs, the different types of IDSs and the detection techniques used in these systems. Finally, the chapter enables the reader to gain the necessary background for understanding the machine learning concepts and the algorithms used in this thesis.

## **Chapter 3: The IPv6 Routing Protocol for LLNS (RPL): Overview, Security Issues, and State-of-the-art Solutions**

This chapter presents an overview of the RPL protocol and a taxonomy of attacks specific to the RPL protocol. In addition, it discusses different defence solutions to tackle RPL attacks while focusing on intrusion detection systems.

## **Chapter 4: RPL's Performance under DIS and SybM Attacks and New Approach for the Intrusions Tolerance**

As a response to the fact that there is a lack of study of the RPL performance under mobile attack, this chapter introduces SybM, a DIS-Sybil attack against RPL with mobile Sybil nodes. The chapter presents an analytical and a simulation-based performance evaluations of RPL under SybM attack and a discussion on how the network performance can be affected. Furthermore, the chapter describes and assesses a novel approach, namely RPL-MRC, to improve the RPL's resilience to both DIS and SybM intrusions.

## **Chapter 5: Intrusion Detection and Tolerance Systems for RPL's Security**

This chapter introduces two IDS solutions to detect and tolerate attacks against RPL networks. The first approach, named T-IDS, is a specification-based cross-layer trust-based IDS that copes with the RPL's security issues related to the lack of mobility and identity management mechanisms for RPL. The second approach that represents our main contribution is named RF-IDSR. It is an anomaly-based intrusion detection and tolerance system that uses machine learning techniques.

## **Chapter 6: Conclusions and Perspectives**

This chapter concludes the dissertation by summarising the key contributions and discussing some directions for future work.

# Chapter 2

## Background: The Internet of Things, Intrusion Detection Systems, and Machine Learning

This chapter presents a thorough background on the IoT paradigm and its security issues. Besides, it provides the preliminary information about the definitions relevant to IDSs, the different types of IDSs and the detection techniques used in these systems. Finally, the chapter enables the reader to gain the necessary background for understanding the machine learning principle.

### 2.1 The Internet of Things (IoT)

#### 2.1.1 IoT Definition

The Internet of Things (IoT) concept was coined in 1999 by Kevin Ashton. The basic idea is that smart, low-power and low-processing objects (things) are able to interconnect, interact, cooperate with each other, and transfer sensing data to the Internet using compatible and heterogeneous wireless technologies, where computing and communication systems are seamlessly embedded [31]. Thus, any electronic device and anything such as mobile devices, home objects (e.g., fridges, dish washers, etc.), temperature control devices, cloth, food, animals and trees are now equipped with sensing, communication, computing and/or processing capabilities. The fact of building a digital counterpart to any entity and/or phenomena in the physical realm enables IoT objects to communicate and interact via wireless technologies such as RFID (Radio Frequency IDentification), ZigBee, WSN (Wireless sensor network), WLAN (wireless local area network), NFC (Near Field Communication), DSL (Digital Subscriber Line), GPRS (General Packet Radio Service), LTE (Long Term Evolution), Bluetooth, or 3G/4G [32][12].



Indeed, there exist different IoT definitions and architectures provided by various standards and industrial organisations. The Institute of Electrical and Electronics Engineers (IEEE) defines the IoT as a collection of items with sensors that form a network connected to the Internet [33][34]. The European Telecommunications Standards Institute (ETSI) uses "machine-to-machine (M2M)" rather than using the expression "Internet of Things". ETSI defines M2M communications as an automated communications system that makes decisions and processes data operations without direct human intervention [35]. Cisco organisation, uses the expression "Internet of Everything (IoE)" and defines it as a network that consists of people, data, things, and processes, where information and actions are created in and moved through this network [36].

This last decade, several IoT architectures have been proposed in the literature [3]. For instance, the middleware-based architecture, the Service Oriented Architecture (SOA-based) and the five-layer architecture. This latter divides the structure of IoT into five layers: the perception, sensing or device layer, the network, transmission or communication layer, the middleware/service management layer, the application layer, and the business layer as illustrated in Figure 2.1 [32]. Nevertheless, from the pool of the proposed architectures, the basic (general) common model is the one in Figure 2.2, known as the three-layer architecture consisting of the perception layer, the network layer, and the application layer [3][12][32].

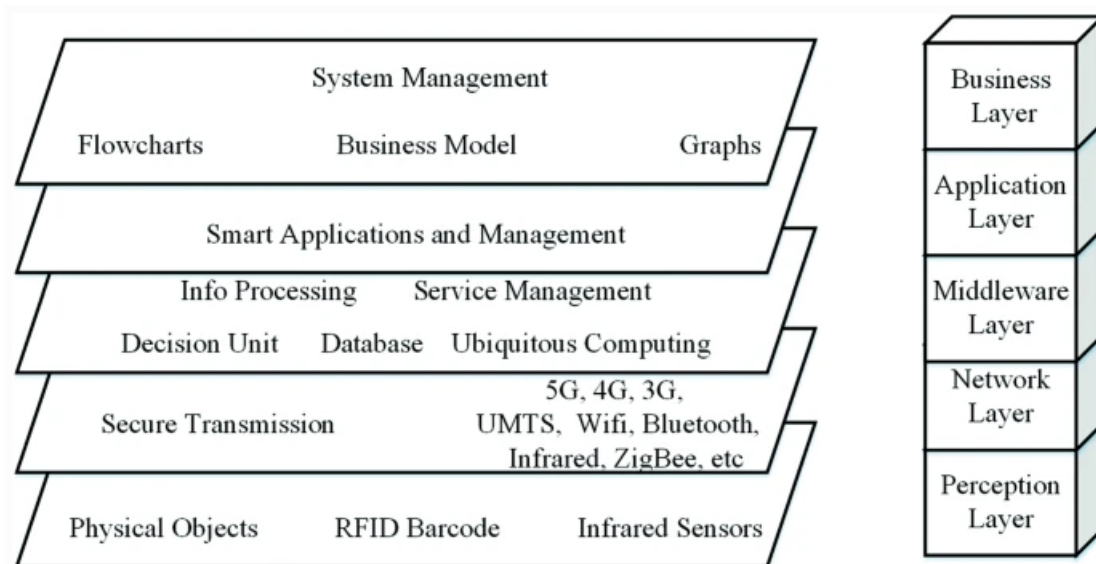


Figure 2.1: IoT five-layer architecture [1].

### 2.1.1.1 Perception/Sensing Layer

It is the core layer of IoT as it allows perceiving and collecting all kinds of information from the physical world, using technologies such as sensors, WSN, tags and reader-

writers, RFID system, camera, global position system (GPS), objects, and so on.

### 2.1.1.2 Network Layer

This layer is also called transport layer as it provides transparent data transmission capability. It is an intermediary layer that provides an efficient, reliable, trusted network infrastructure platform to send information from the perception layer to the upper layer and large scale industry application using existing mobile communication network, radio access network, WSN and other communications equipment, such as GSM (global system for mobile communications), GPRS, WiMax (worldwide interoperability for microwave access), WiFi, Ethernet, etc.

### 2.1.1.3 Application/Service Layer

This layer relies on SOA, cloud computing , and other technologies to process complex data and uncertain information, such as restructuring, cleaning and combining, and provides directory service, market to market (M2M) service, Quality of Service (QoS), facility management, etc. Furthermore, the application transforms information to content and provides good user interface for upper level enterprise application and end users, such as logistics and supply, disaster warning, environmental monitoring, agricultural management, production management, and so on.

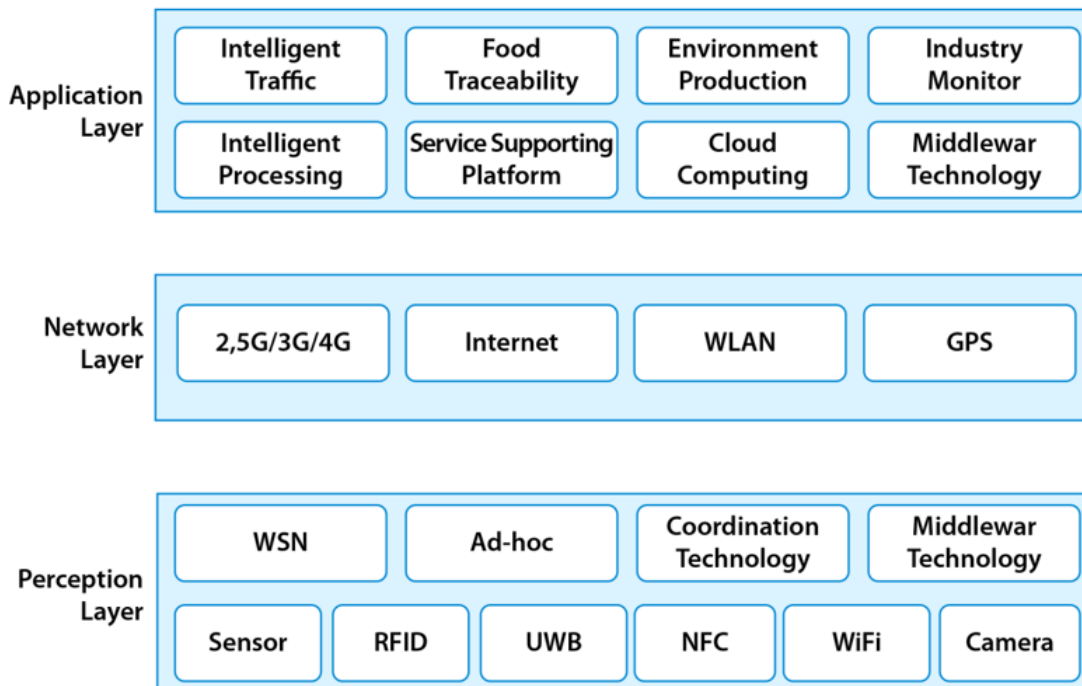


Figure 2.2: IoT three-layer architecture.

### 2.1.2 IoT's Applications

One of the main building blocks of the IoT are the Low-power and Lossy Networks (LLNs). LLNs are made of a collection of interconnected embedded resource-constrained devices, such as RFID and sensor nodes with low computational and storage capabilities and are often battery operated. The LLNs-IoT has a great impact on several aspects of everyday business and personal lives, where sensor measurements can be read, processed, and analysed. Indeed, IoT applications serve different users needs in different contexts, as illustrated in Figure 2.3. Applications for personal lives range from advanced health monitoring, smart leaving, enhanced learning, to improved security. For example, in an e-health application, a patient (inside or outside the hospital) wears a heart rate monitor, wrapped around the chest or a smart watch on the wrist, which is continuously reading and transmitting the heart rate sensor readings to another IoT node. Hence, the doctor can monitor conditions of his patients in real-time, and thus, emergencies can be handled on the fly. In a smart home, smart refrigerators can display information on ingredients to buy or to throw away. Windows, doors and cameras can signal intrusion. Smart televisions enable users to surf the Internet, make purchases, and share photos. Also lights, heaters, air conditioners, and washing machines can be manipulated remotely [3][12][31][37].

From another side, applications for business include smart cities and energy, smart environment, smart industry and Industry 4.0 [38][39], smart health and smart agriculture. Applications can be smarter energy management systems (smart grid) to monitor and manage energy consumption. Smart surveillance to ensure safety. Automated transportation by introducing smart roads. Vehicular and Industrial automation (e.g., predictions on equipment malfunction). Environmental monitoring such as water quality monitoring and water distribution, air pollution monitoring and fire detection. For smart tracking in supply chain management, IoT technologies such as RFID tags can be used for tracking objects from production, all the way to transportation. Besides, in green houses, micro-climate conditions are controlled to maximize the production and the quality of products. Also, in smart grid, efficient energy consumption can be achieved through continuous monitoring of electric consumption. In fact, smart home and e-health are the biggest potential markets for IoT networks [3][12][31].

### 2.1.3 LLNs-IoT Characteristics

In the literature, there is no common definition of IoT. It is highly related to the vision of each academic or business entity [40]. To get the IoT concept closer to reality, the following characteristics need to be addressed

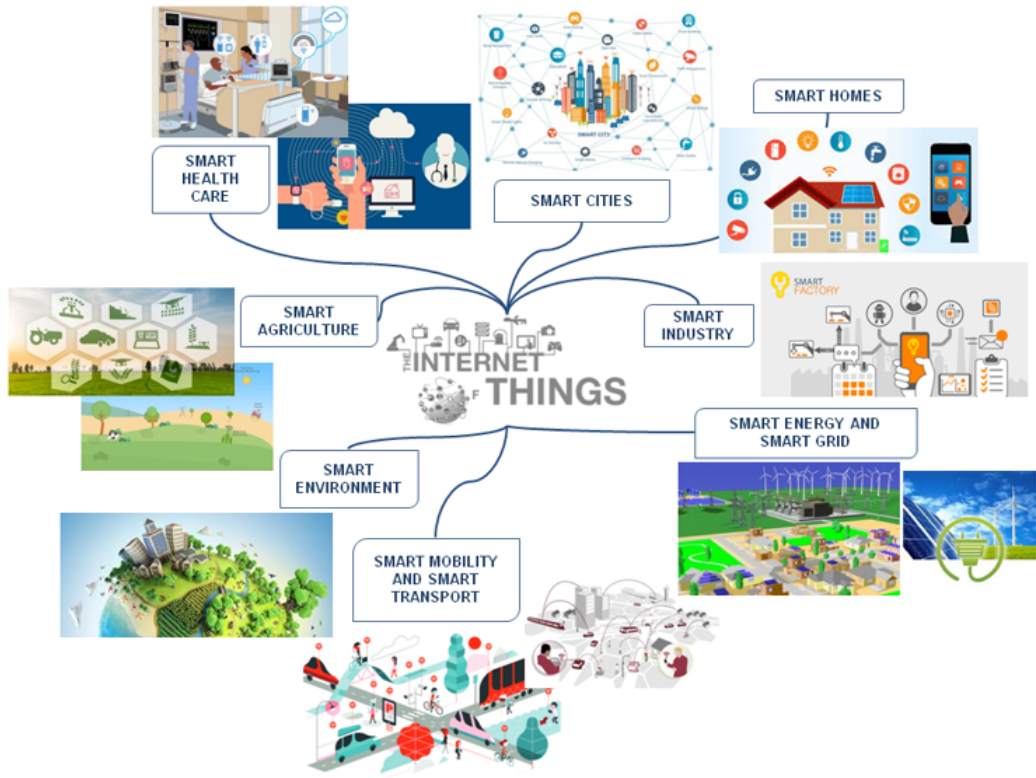


Figure 2.3: IoT application domains.

- **Heterogeneity.** One important characteristic of IoT is the large heterogeneity of devices and technologies taking part in the system. The IoT's devices are deployed using several hardware platforms and have different computational and communication capabilities.
- **Scalability.** Since anything can be connected, the number of entities participating in the network will dramatically increase. In this context, scalability should be considered for identification and addressing, communication and networking, data and information management, and security management.
- **Connectivity and Ubiquity.** The ubiquitous connectivity to the Internet in IoT enables objects to access the network and exchange data using the wireless medium. Nonetheless, these characteristics rise several challenging problems that need to be addressed such as security and QoS.
- **Self-organisation and Self-healing.** Since the number of IoT objects and their connection and location states change dynamically, IoT smart objects are equipped with embedded intelligence allowing them to autonomously react and self-organise themselves into transient ad hoc networks according to specific situations, states, and to the current context. Intruders can exploit these characteristic to trigger several attacks against IoT networks.

- **Resources limitation.** As IoT objects are characterised by their heterogeneity, they are different in term of energy, storage and computation capabilities. Therefore, IoT platforms development needs to optimise and minimise the objects energy, storage and computation usage as much as possible.
- **Interoperability.** In realm of IoT there are many largely distributed and heterogeneous objects and things, with different power, processing and storage capacities, that co-existent and need to communicate and cooperate in order to achieve common goals. This heterogeneity and diversity in terms of capacities, vendors/manufactures and services increases the need of conceiving systems and protocols able to work in an interoperable way. In this context, standards are particularly important.
- **Mobility.** Most of the smart devices and IoT actors are mobile. This characteristic causes several changing to the network conditions, which makes it difficult to communicate with each other. Furthermore, not handling mobility can generate more security breach.
- **Security and Privacy.** As IoT is going to affect every aspect of human and business lives, and as IoT devices generate a huge amount of data, IoT entities should be equipped with strong security and privacy policies. This includes securing the devices themselves (i.e., hardware), exchanged data and information, communications and networks, and endpoints. Several complex security mechanisms and protocols exist for different networks; however, because of the constrained nature of IoT devices and networks only lightweight protocols can be used to keep the balance between maximizing security and minimizing resource consumptions.

## 2.2 Internet of Things Protocols

In IoT, different entities communicate over the network using a diverse set of protocols and standards. Obviously, there is no one technology that is able to cover all use cases. Different groups such as the Internet Engineering Task Force (IETF) worked and are still working to provide and standardise protocols in support of LLNS in the context of IoT. The following subsections outline some standards underpinning LLNs in the field of IoT.

### 2.2.1 IEEE 802.15.4

The IEEE 802.15.4 standard defines low-power wireless embedded radio communications at 2.4 GHz, 915 MHz and 868 MHz. It specifies a sub-layer for Medium

Access Control (MAC) and a physical layer (PHY) for low-rate wireless private area networks (LR-WPAN). Due to the low power consumption, low data rate, low cost, and high message throughput specifications, the IEEE 802.15.4 standard is used by several IoT protocols such as ZigBee, Wireless HART, MiWi, ISA 100.11a, and 6LoWPAN. IEEE 802.15.4 has been enhanced to be the most common 2.4 GHz wireless technology to address the emerging needs of embedded networking applications. The first version of the standard was released in 2003, and was then revised in 2006. This later was then revised in 2011 and then in 2015. IEEE 802.15.4 at 2.4 GHz is used almost exclusively today as it provides reasonable data rates, reliable communication, can handle a large number of nodes, and can be used globally. IEEE 802.15.4 networks can form various topologies, such as star, cluster-tree or mesh (peer-to-peer). The IEEE 802.15.4 standard defines two kinds of devices in the network: Full Function Devices (FFD) and Reduced Function Devices (RFD). The FFD may function as a common node or it can serve as the coordinator of a Personal Area Network (PAN or PAN Coordinator). The FFD is an extremely simple device with very modest resource and communication requirements, and thus cannot act as coordinator. The maximum transmission unit is 127 bytes [2][14][15].

### 2.2.2 IPv6 Over LoW Power Wireless Area Networks (6LoWPAN)

Given the potentially huge number of connected objects, IPv4 cannot be used because of its limited address space. Thus, a much better choice is to use IPv6 with its 128-bit addresses and its ability to allow network auto-configuration and stateless operation. Using IPv6, every smart object can be connected to other IP-based networks, without the need for gateways or proxies. Hence, objects can define their addresses in very autonomous manner. This enables to reduce drastically the configuration effort and cost. Because of the limited packet size, the low power capacity, and other constraints of IoT, the research community (6LoWPAN IETF Working group) developed a compressed version of IPv6, named 6LoWPAN (IPv6 over LoWpower Wireless Area Networks) [13]. It is a simple and efficient mechanism to shorten the IPv6 address size for constrained devices, while border routers can translate those compressed addresses into regular IPv6 addresses. The 6LoWPAN standard allows the extension of IPv6 into the wireless embedded environment. Due to the resource constrained nature of the devices or things, 6LoWPAN network use compressed IPv6 protocol for networking and mostly use IEEE 802.15.4 as data-link and physical layer protocol. 6LoWPAN defines an adaptation layer between IPv6 and IEEE 802.15.4 (MAC/PHY layer). It defines IPv6 header compression by removing a lot of IPv6 overheads, and specifies how packets are routed in wireless

networks that use the IEEE 802.15.4 protocol. It also defines fragmentation of IPv6 datagram when the size of the datagram is more than the IEEE 802.15.4 Maximum Transmission Unit (MTU) of 127 bytes [13]. Figure 2.4 and Figure 2.5 show the 6LoWPAN architecture and protocol stack, respectively.

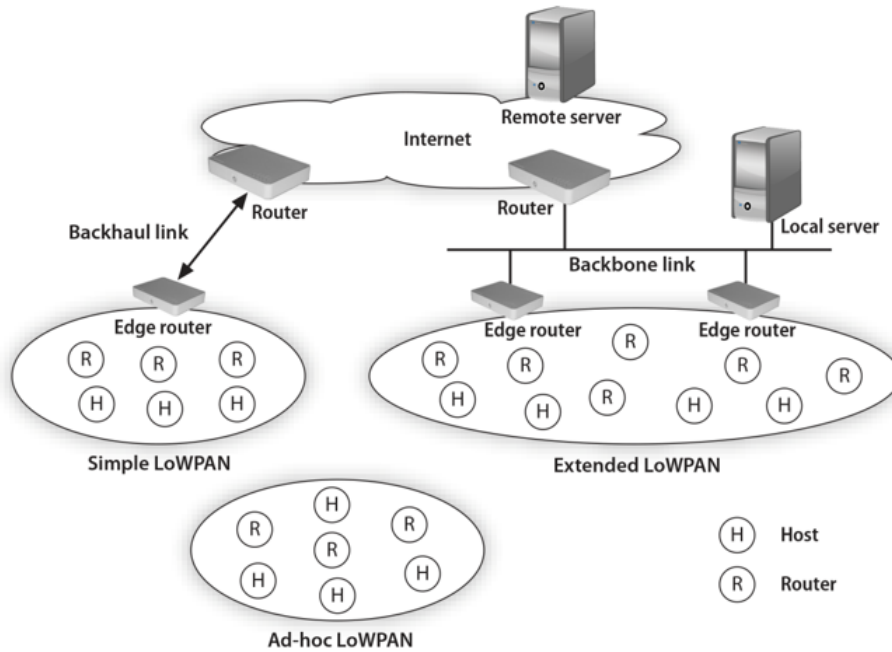


Figure 2.4: 6LoWPAN architecture [2].

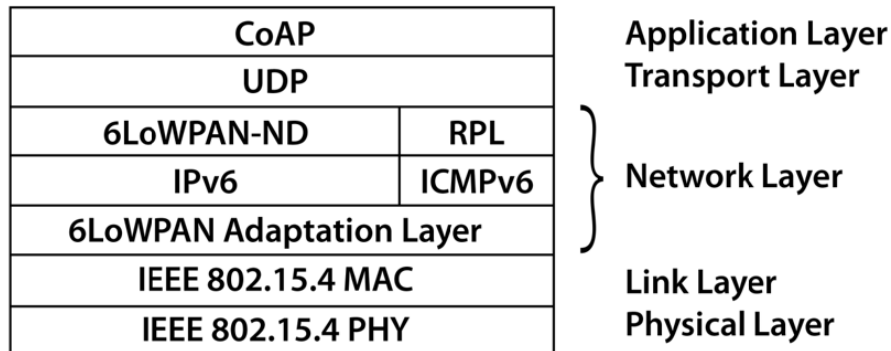


Figure 2.5: 6LoWPAN protocol stack.

### 2.2.3 Routing Protocol for Low-Power and Lossy Networks (RPL)

The Routing Protocol for Low-Power and Lossy Networks (RPL) is the first standardised routing protocol mainly targeting 6LoWPAN. Nevertheless, RPL is not restricted to use with 6LoWPAN as it provides solutions for low-power, wireless,

and unreliable networks such as LLNs. RPL deals with the constrained nature of such networks by considering limitations both in energy power and computational capabilities. Besides, RPL operates within the LLN (6LoWPAN) domain, and terminates at the border router. An in-depth overview of RPL is presented in Chapter 3.

## 2.2.4 Constrained Application Protocol (CoAP)

In IoT networks, the connection-less User Datagram Protocol (UDP) is mostly used as the transport layer. This is due to the fact that it is hard to maintain a continuous connection between low-powered devices using lossy links. In these circumstances, the Constrained Application Protocol (CoAP) was proposed by the IETF Constrained RESTful Environments (CoRE) working group. CoAP is an application layer protocol for IoT that modifies some HTTP functionalities to meet low power consumption and lossy and noisy links characteristics of IoT. CoAP protocol is a standardised, lightweight and efficient web transfer protocol specifically designed for low-power networks, with high packet error rates and relatively small throughput, such as 6LoWPAN networks. It works on constrained devices on top of the unreliable UDP transport layer to provide good interface for the standard Internet services. When CoAP is used with 6LoWPAN as defined in RFC4944 [14], messages fit into a single IEEE 802.15.4 frame to minimise fragmentation. By introducing CoAP, application layer and applications themselves do not need to be re-engineered to run over low-power embedded networks. This is because CoAP protocol implements a set of techniques to compress application layer protocol metadata without compromising application interoperability, in conformance with the REpresentational State Transfer (REST) architecture of the web. Figure 2.6 demonstrates the overall functionality of CoAP protocol [3][41].

With the introduction of CoAP, a complete networking stack of open standard protocols that are suitable for constrained devices and environments become available [42]. Furthermore, since CoAP is used in the IoT as an application protocol then end-to-end security between two applications can be provided with the Datagram Transport Layer Security (DTLS). The secure version of CoAP is CoAPs that uses compressed version of DTLS to protect CoAP messages between two applications in the IoT. Reliability in CoAP is achieved through the use of acknowledgements messages [43][44].



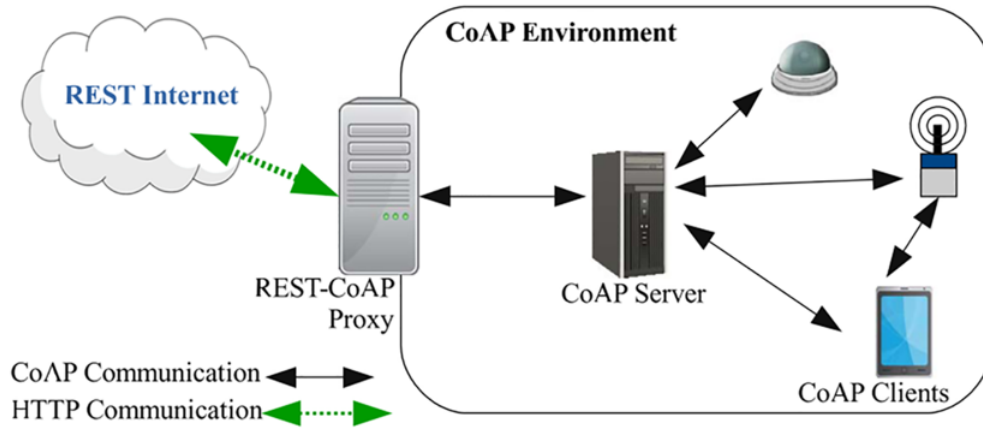


Figure 2.6: CoAP functioning within an IoT environment [3].

## 2.3 Security Challenges in IoT

The security of IoT systems is a serious issue due to the resource limitation, mobility and identification gaps, and easy capture of IoT's devices. In addition, because more devices/things become connected every day and more smart devices are installed in homes, hospitals and building, the number of vulnerabilities an intruder could use to compromise IoT networks increase continuously.

Furthermore, the 6LoWPAN relaying on IEEE 802.15.4 and IPv6 causes vulnerabilities and creates several new threats from the two sides, thus targeting the different layers of an IoT architecture ranging from the application/service layer to the perception physical layer. According to [3], the three-layer architecture borrows layers and concepts from the network stacks and thus their respective threats such as the unauthorised access to data and DoS or availability attacks.

Nawir et al. [45] proposed a taxonomy of attacks on IoT as follows: (1) device property (low-end device class, high-end device class), (2) access level (passive, active), (3) adversary location (internal, external), (4) strategy (physical attacks, logical attacks), (5) information damage level (interruption, eavesdropping, alteration, fabrication, message replay, MITM), (6) host-based (user, software, hardware), (7) protocol-based (deviation from protocol attacks, protocol disruption attacks), and finally, (8) communication stack protocol (layer-based: physical, link, network, transport, and application). From another hand, the authors in [4] classified the IoT's threats by design challenges according to the IoT's characteristics, as in Figure 2.7. Indeed, the security challenges in IoT systems are related to security issues arising from the different IoT layers. In the following subsections, we present attacks corresponding to each layer, while focussing on the network layer.

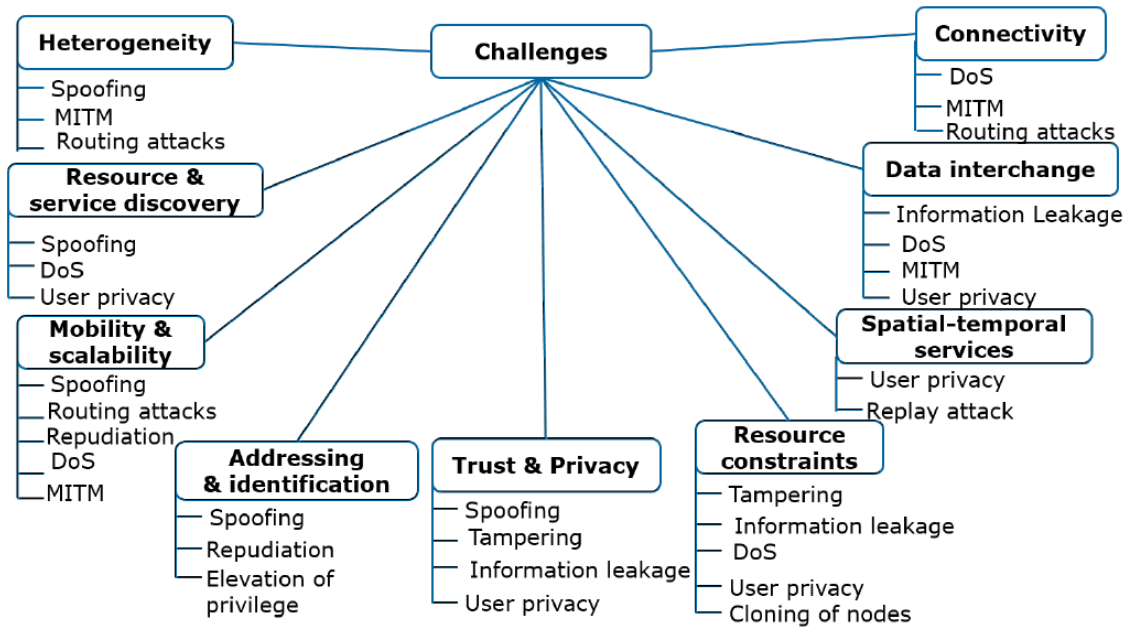


Figure 2.7: IoT security challenges according to [4].

### 2.3.1 Perception/Sensing Layer

The perception layer also called sensing or even device layer represents the physical world where heterogeneous, resource constrained and highly distributed IoT devices co-exist. In this layer data/information is sensed, collected and then sent to the upper layers for further processing. Different technologies and protocols can be used in the perception layer, such as RFID technology, Wi-Fi, Long Term Evolution (LTE), WiMax, Near Field Communication (NFC), Bluetooth, and ZigBee, where communication channels are extra susceptible to several breach and attacks. An intruder can easily access physical sensing devices in order to damage or reprogram illegal actions on them.

Besides unauthorized access to data, DoS and DDoS attacks, several other attacks can be triggered against the sensing layer depending on the communication technology [45][46].

#### 2.3.1.1 Side-Channel Attacks

Any attack based on information gained from the physical implementation of a cryptosystem. It consists on the evaluation of leakage information that emanates from a physical implementation to recover the key the device is using. These could be timing or power traces of inner operations of the device, or faulty outputs produced by it. Because of the increased openness in IoT, different possible side channel attacks can be triggered such as: timing attacks, power analysis attacks, electromagnetic analysis attacks, fault induction attacks, optical side channel attacks, traffic analysis attack, acoustic attacks, and thermal imaging attacks [47].

### **2.3.1.2 Brute Force Attack**

It is a cryptanalytic attack that can be used to attempt to decrypt any encrypted data. Ling et al. [48] demonstrated that some smart home plugs are very vulnerable to this attack.

### **2.3.1.3 Man-In-The-Middle (MITM) Attacks**

In these attacks, an attacker is looking to interrupt and breach communications between two things. Thus, two parties believe that they are communicating directly with each other when they are not. Actually, the attacker secretly intercepts and transmits messages between the two entities tricking them into thinking they are still getting legitimate messages. Many cases have already been reported within this threat. Authors in [49] found that IoT hubs can be attacked using simple MITM attacks. Griffin [50] proposed biometric-based cryptographic techniques to counter this kind of attacks.

### **2.3.1.4 Unfairness Attacks**

Malicious nodes misbehave and break the standard communication rules of the IoT IEEE 802.15.4 MAC layer to capture the channel with higher priority utilisation. Thus attackers get a dominating position and hold unfair advantages over the other nodes. In [51], the authors presented new algorithms to counter the GTS (Guaranteed Time Slots) MAC unfairness attack.

### **2.3.1.5 Masquerading Attacks**

The attacker steals and tries to use the identity of the authorised node in the network. Actually, this attack can be triggered at the network layer (Sybil and/or ClonID attacks).

### **2.3.1.6 Other Attacks**

There are several other attacks such as, node tampering and physical attacks, collision attacks (e.g., back-off manipulation attack), jamming attacks, battery exhaustion attacks, replay attacks, traffic sniffing/eavesdropping attacks, data-corruption/message-alteration attacks, key sniffing attacks, proof-of-concept attacks, tag modification and tag cloning attacks, RFID authentication attacks, and so on.

## **2.3.2 Network Layer**

Almost known as Wireless Sensor Networks (WSNs), the network layer represents an intermediate layer that is used to aggregate and transmit sensed data from the

perception layer to the application layer using existing wired and wireless communication networks like the Local Area Network (LAN) such as WiFi, the Personal Area Network (PAN) such as ZigBee, or even the Wide Area Network (WAN) such as LTE and GPRS (see Figure 2.2). According to Kumar and Patel [52], it is the "Central Nervous System" of the IoT. WSNs are the most popular networks for IoT regarding their ability to cover large areas of things and to retain adequate consumption of energy. It is important to pinpoint that the network layer should support the communication requirement for latency, bandwidth and security. Nevertheless, the characteristics of IoT environments cause several security and privacy concerns, especially associated to the network layer. It should be pointed out that in 6LoWPAN networks, this layer is composed of two sub-layers: the 6LoWPAN adaptation layer and the routing protocol, probably RPL. The fact that RPL supports MP2P and P2MP traffic patterns makes 6LoWPAN networks more vulnerable to routing attacks. A variety of attacks targeting the two sub-layers of the network layer have been identified in the following subsections.

### 2.3.2.1 Adaptation Layer Attacks

The adaptation layer is implemented at the border router for translating packets between the 6LoWPAN network and Internet. The adaptation layer is mandatory as the size of IEEE 802.15.4 frames (limited to 127 bytes) do not permit to use conventional IPv6 packets (1280 bytes). In this condition, header compression, fragmentation and reassembling are handled by the adaptation layer. The authors in [25] presented a survey on 6LoWPAN related attacks.

**2.3.2.1.1 Fragmentation Attacks** The border router is normally a wired node and has strong security protection. However, the packet fragmentation and reassembly progress still have some vulnerabilities. In these attacks the attacker can modify or reconstruct the packet fragmentation fields like datagram size, datagram tag or datagram offset. These attacks can cause critical damage to a node, for instance, reassemble buffer overflow because of packet re-sequence, exhausting resource because of processing unnecessary fragmentation, or shutting down and rebooting. Also, as there is no authentication mechanism at the receiver side for checking that received fragment is not a spoofed or duplicated one, the attacker may put his own fragments in the fragmentation chain. The fact that integrity checksums and signatures are calculated over whole packets instead of over intermediate fragments, the validity of the fragmented packets cannot be verified before packet reassembly. As a consequence, an attacker can fill up the limited buffer space of IoT devices with invalid fragments by flooding the resource constrained objects with few large packets [43][25].

**2.3.2.1.2 Authentication Attacks** Unfortunately, there are no mechanisms for 6LoWPAN nodes to authenticate before joining the network. So, it is obvious that malicious nodes can easily join the network and trigger other internal attacks, which is very harmful for IoT applications. Many authentication protocols have been proposed in the literature. The authors in [53] surveyed authentication protocols for IoT according to their mechanisms. The proposed authentication protocols for 6LoWPAN networks are used to check the identity of each device in the network and authenticate it. Providing strong authentication mechanisms can help to mitigate several attacks such as Sybil and CloneID attacks.

**2.3.2.1.3 Confidentiality Attacks** Only legal and authorised nodes can access, watch and control data in the network. Providing confidentiality in 6LoWPAN can help to mitigate various attacks such as eavesdropping, MITM, spoofing attacks, and so on. As for authentication, identity management represents a key factor to assure confidentiality [54]. Besides, cryptography is considered the first line for solving the confidentiality and authentication issues. In fact, Internet Protocol Security (IPSec) provides an end-to-end network layer security by enabling the authentication and encryption of exchanged IP packets, using Authentication Header (AH) to provide integrity and authentication, and Encapsulating Security Payload (ESP) headers to provide integrity and confidentiality. Since IPSec is very greedy and supplies energy and space, some research works proposed compressed version of IPsec headers to secure the 6LoWPAN adaptation layer [55][56].

**2.3.2.1.4 Internet Side Attacks** 6LoWPAN IoT devices and Internet hosts differ strongly regarding their available resources. Indeed, objects in 6LoWPAN networks have limited memory, computational power and very limited security provision. Whereas Internet hosts are equipped with CPUs in the GHz range and huge memory. These capacities differences, in addition to the openness of IoT make 6LoWPAN networks vulnerable to several attacks from Internet. For avoiding such attacks a firewall could be installed on the edge router (see Figure 2.4) to control the malicious packets from Internet. An adaptation layer-based approach has been proposed for enabling security bootstrapping between the IoT domain and the Internet with existing IP security protocols such as DTLS [43].

The authors in [25] classified authentication, confidentiality, and Internet attacks as adaptation layer attacks; however, we believe that these attacks can target any layer of the three-layer architecture.

### 2.3.2.2 Routing Attacks: RPL Threats

As RPL's security is the main focus of our contributions, we present an in-depth analysis of the protocol and its vulnerabilities in Chapter 3.

### 2.3.3 Application/Service Layer

The application layer consists on business solutions that allow the final users to interact remotely with the physical world composed of things, devices and people. Furthermore, it provides services to manage, analyse and visualise measurements and outputs using users specific interfaces. It defines various applications in which the IoT can be deployed, for instance, smart health, smart cities, smart environment and smart home. Furthermore, it allows building business models, graphs, flow-charts based on the received data. Because IoT devices have specific characteristics, instead of using HTTP, lightweight protocols such as CoAP has been developed to support application layer communications. As IoT network is directly connected to the unsecured Internet, it can undergo attacks from it (Internet) such as transactions replays, traffic congestion generation, and DoS and DDOS attacks. Indeed, attackers may trigger the overwhelm attack to destroy the routing by generating huge traffic to the edge router, and the path-based DoS attack to deplete resources by injecting false messages [57]. For avoiding application-layer attacks, a firewall could be installed on the edge router to control the malicious packets from Internet. From another hand, Datagram Transport Layer Security (DTLS) may provide end-to-end security since it represents a solution to confidentiality, integrity, authentication and non-repudiation security problems for application layer communications using CoAP [55][58]. Additionally, replay attacks may be mitigated with DTLS, using a different nonce value for each secured CoAP packet [59].

## 2.4 Intrusion Detection Systems (IDSs)

There are vulnerabilities from inside the network and from the Internet that go beyond the encryption and authentication first lines of defence for the IoT communications. Thus, IDSs are required as a second line of defence. An IDS is a powerful tool for collecting, monitoring and analysing user information, networks, and services to identify and protect against intrusions that threaten the confidentiality, integrity, and availability of an information system [60]. The operations of an IDS can be summarised on three stages: i) The monitoring stage, which relies on network-based or host-based sensors. ii) The analysis stage, which uses feature extraction or pattern identification methods. iii) The detection stage, which relies on methods for intrusion detection.

### 2.4.1 IDSs Types

There are two types of IDSs: i) A host-based IDS (HIDS) is designed to be implemented on a single system and to protect that system from intrusions or malicious attacks, which will harm its operating system or data. ii) A network-based IDS (NIDS) sniffs network traffic to detect intrusions and malicious attacks. NIDS<sup>1</sup> are primarily used to counter attacks against the network. They analyse activities and nodes behaviours in the network and aim to detect intruders that are trying to disrupt the network. IDSs can be classified depending on their location or on the method used for detection.

### 2.4.2 IDSs Locations and Methods

#### 2.4.2.1 IDSs over Networks

**2.4.2.1.1 Centralised (Monolithic)** In this approach, each node monitors the operation of all of its neighbours in the network and transfer collected information to a central intrusion detection node to be analysed. It suffers from the following deficiencies:

- Scalability: It is difficult to guarantee scalability as a network size grows. Furthermore, because lots of information need to be transferred from monitoring nodes to a central node, this creates heavy overhead, which causes severe degradation of the network performance.
- Robustness: The central node represents a single point of failure making the overall IDS crippled in case of its failure or if it is attacked.

**2.4.2.1.2 Distributed (Cooperative)/Host-based** In this approach, the IDS is placed on each node within the network. Each host-based IDS monitors only a small portion of the network (neighbours). The distributed host-based IDSs cooperate to analyse and detect intruders. They can make a coherent inference and make a global decision. Monitoring host-based nodes are called watchdogs.

- It is the most used approach in the literature because it resolves most of security breach in the network. Nevertheless, it is hard to repair and maintain the overall system in addition to the generated overhead on the monitored parts of the network. Furthermore, it requires a lot of memory and calculation resources.

---

<sup>1</sup>In this thesis, we use IDS instead of NIDS, for short.

**2.4.2.1.3 Hierarchical/Hybrid** In this approach the network is divided on a number of hierarchical monitoring areas, where each IDS monitors a single area. Instead of transferring all the collected information from monitoring nodes to a central IDS, each single IDS at each level of monitoring area performs local analysis and sends its local analysis results up to the IDS at the next level in the hierarchy. Thus, IDSs at higher levels only need to analyse transferred local reports collectively.

- Scalability: Shows better scalability by allowing local analyses at distributed local monitoring areas.
- Robustness: If the topology of the network changed the network hierarchy changes as well and the whole mechanisms to aggregate monitored information must be changed. Furthermore, when a monitoring node residing at the highest level is attacked or fails, the sub-network related to this node easily escape detection.

#### **2.4.2.2 IDSs over Detection Method**

**2.4.2.2.1 Signature-based** This IDS is also known as misuse-based IDS. In this IDS, signatures of malicious activities or codes are stored in a database or a list. The signature patterns in packets are matched with the stored ones. If match founds then the IDS raises alarm for the attack. The IDS compares the current activities in a network or in a device against predefined and stored attack patterns (signatures). This approach cannot detect new attacks, needs specific knowledge of each attack, has a significant storage cost that grows with the number of attacks, and has a high false negative but low false positive rate.

**2.4.2.2.2 Anomaly-based** This type of IDS determines the ordinary behaviour of a network or a device, uses it as a baseline, and detects anomalies when there are deviations from the baseline. This approach can detect new attacks but has comparatively high false positive and false negative rates because it may raise false alarms and/or cannot detect attack when attacks only show small deviations from the baseline. There exist various anomaly-based detection techniques that have been summarised in Figure 2.8 [1].

**2.4.2.2.3 Event-based** In this IDS system, the malicious events patterns are defined, a priori, and stored in a database. Thus, the event based IDS captures the events triggered in the network and analyses them. If an event is suspicious, the IDS raises alarm for attack detection.



Technique	Advantages	Disadvantages
Data mining	1- Models are created automatically	1- Based on historical data
	2- Applicable in different environments	2- Depends on complex algorithms
	3- Suitable for online datasets	
Machine learning	1- High detection accuracy	1- Requires training data
	2- Suitable for massive data volumes	2- Long training time
Statistical model	1- Suitable for online datasets	1- Based on historical behavior
	2- System simplicity	2- Detection accuracy depends on statistical and mathematical operations
Rule model	1- Suitable for online datasets	1- Based on a set of rules
	2- System simplicity	2- High false positive rate
Payload model	1- High detection accuracy for known attacks	1- Privacy issues
		2- Long processing time
Protocol model	1- High detection accuracy for a specific type of attack	1- Designed for a specific type of protocol
Signal processing model	1- High detection accuracy	1- Depends on complex pattern-recognition methods
	2- Low false positive rate	

Figure 2.8: Advantages and disadvantages of anomaly-based detection techniques [1].

**2.4.2.2.4 Specification-based** This type of IDS is also known as software engineering based IDS or Finite State Machine (FSM) based IDS. It defines a typical behaviour of the protocol (e.g., a routing protocol as an FSM) and uses it as a baseline for detecting abnormal behaviour. If an abnormal behaviour is detected, the IDS raises alarm.

**2.4.2.2.5 Hybrid-based** This type of IDS combines the types above to get better detection results with less negative impacts on the network performances.

## 2.5 Machine Learning Concept

Machine Learning (ML) is a subset of Artificial Intelligence (AI). ML has emerged as the method of choice for developing practical solutions in many areas of technology and science such as, text or document classification, computer vision (e.g., objects recognition and identification, face detection, etc.), speech processing (e.g., speech recognition and synthesis, speaker verification and identification, etc.), natural language processing (e.g., part-of-speech tagging, context-free parsing, etc.), robotics and autonomous vehicle control, and other applications like fraud detection for credit card and intrusion detection. Indeed, application areas of ML keep expanding as most prediction problems can be cast as learning problems [61].

### 2.5.1 ML Common Terminology

There are a terminology that is commonly used in ML [61].

- **Items** are examples, samples, or instances of labelled or unlabelled data used for learning or testing (evaluation).

- **Features** are the set of attributes associated to an item. The features are often represented as a vector.
- **Labels** are values or categories assigned to the items. For instance, in classification problems, items are assigned specific classes (categories) such as malicious and normal.
- **Training (Fitting) sample** is the set of items used to train (fit) the ML algorithm. In a classification problem, the training data consist of a set of items with their associated labels.
- **Testing sample** is the set of items used to evaluate the performance of the ML algorithm. The testing sample is different from the training sample as it represents the unseen data.

### 2.5.2 ML Tasks

The main objectives of ML consist of generating accurate predictions for unseen items and of designing efficient and robust algorithms to produce these predictions. ML techniques are used for several tasks as follows [61].

- **Classification** consists of assigning a category for each item. For instance, classifying a document as business, politics, sports, etc. Another example is classifying traffic packets as malicious or normal.
- **Regression** consists of predicting a real value for each item, such as the prediction of stock values, variation of economic variables, etc.
- **Ranking** consists of learning to order items according to some criterion, such as Web search.
- **Clustering** consists of partitioning a set of items into homogeneous subsets. An example is identifying natural communities within large groups of people in the context of social network analysis.
- **Dimensionality Reduction** consists of transforming an initial representation of items into a lower-dimensional representation while preserving some properties of the initial representation. An example is preprocessing digital images in computer vision tasks.

### 2.5.3 ML Algorithms

There exist four groups of ML algorithms: supervised, unsupervised, semi-supervised, and reinforcement learnings. In this chapter, we focus on the supervised algorithms,

which consist of a training phase and a testing phase. At the training phase, the algorithms learn the relationship between the input values (i.e., training sample) and the labels (e.g., the label 0 for normal behaviour, and the label 1 for malicious behaviour). At the testing phase, the algorithms try to predict the output values (i.e., classes, labels) of the testing sample.

Since the aim of an IDS is to decide whether a packet either belongs to normal or attack traffic, intrusion detection is considered as a classification problem. Thus, IDS implementation can be based on different ML classifiers. Indeed, ML-based IDS can change its execution strategy as it acquires new information. This property makes ML desirable to use for any situation. In the following, we present the ML classifiers and the Deep Learning (DL) model investigated in this study. The ML classifiers represent the frequently applied algorithm for intrusion detection. DL is a particular technique of ML that groups generic algorithms mimicking the biological functioning of a brain [62], as it is presented in Section 2.5.3.7.

### 2.5.3.1 Decision Tree (DT)

DT classifier represents the standard for partition-based models. DT main idea is to “break up a complex decision to into a union of several simpler decisions, hoping the final solution obtained would resemble the intended desired solution” [63]. Hence, DT splits data into many branch-like segments such as in the tree structure, leaves represent classifications also known as labels, intermediate nodes represent features, and branches are conjunctions of features that lead to classifications, as depicted in Figure 2.9.

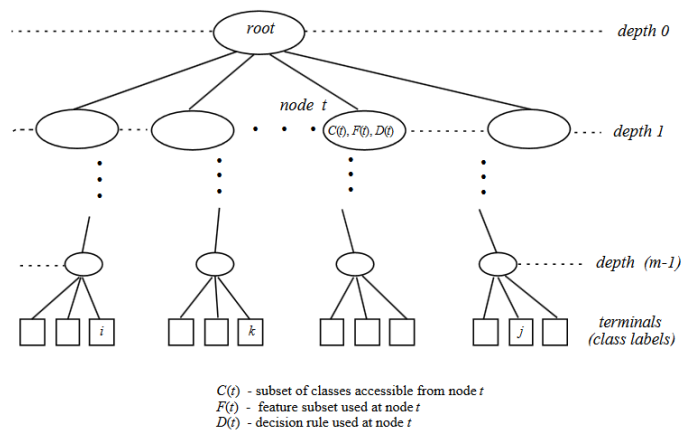


Figure 2.9: Decision tree technique illustration.

### 2.5.3.2 Random Forests (RF)

RF classifier [64] constructs a figured large number of uncorrelated DTs, as illustrated in Figure 2.10. Each DT predicts a classification for a sampled input data

from the original dataset. Furthermore, each DT selects a subset of features from the original set for the fullest growing at each node randomly. Finally, RF collects the predictions and selects the most voted one as the final classification. RF is extensively used in data science since it has high accuracy level, speed, and stability, it is easy to parametrise, robust against overfitting, it can be applied to large-scale datasets, and is not sensitive to noise in datasets. RF is also handy for feature selection as it determines the importance of different features during the classification process.

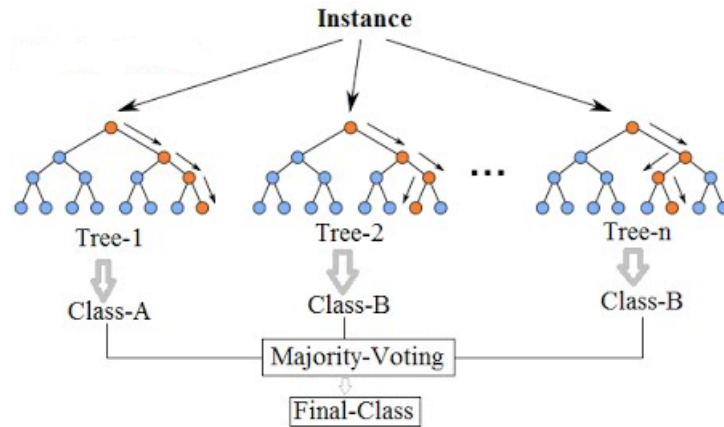


Figure 2.10: Random forests technique.

### 2.5.3.3 K-Nearest Neighbour (KNN)

KNN classifier [65] is a non-parametric supervised ML technique that relies on similarity or distance in feature space to classify samples. In KNN, testing sample (i.e., unlabelled data) is assigned to the class that is most frequently occurred amongst the K nearest neighbours in the training set (see Figure 2.11). The number K, as the square root of the total number of samples in the training dataset. KNN is widely used because it is simple, very scalable, and very fast to converge.

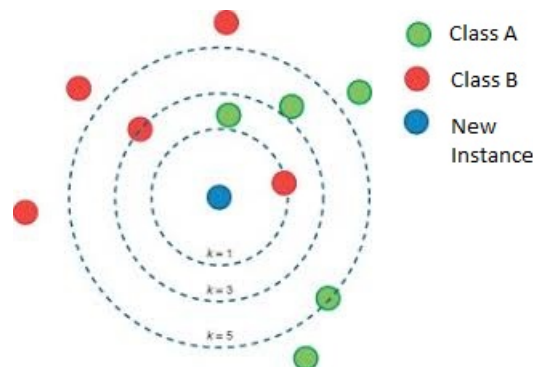
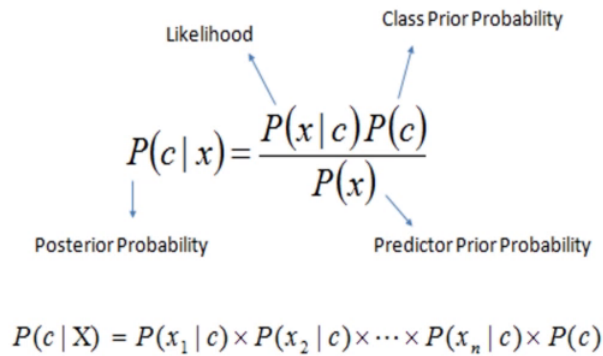


Figure 2.11: K-nearest neighbour (KNN) technique.

### 2.5.3.4 Naïve Bayes (NB)

NB or Bayesian learning [66] is a probabilistic classifier based on probabilities of hypotheses. NB applies Bayes' theorem to calculate probabilities, with the strong assumption that features are independent given class variable, which means that the probability of one feature does not affect the probability of the other one. A prior probability is assigned to each candidate hypothesis based on prior knowledge. NB uses training samples to increase or decrease the probability of a hypothesis to be correct. It classifies a testing sample by assigning the most probable target class, as presented in Figure 2.12.



$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$

Figure 2.12: Naïve Bayes technique.

### 2.5.3.5 Multi-Layer Perceptron (MLP)

MLP Classifier [67] is a feed-forward Artificial Neural Network (ANN) model connecting multiple hidden layers in a directed graph, where each layer is fully connected to the next one. An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a non-linear activation function. MLP employs the back-propagation supervised learning technique for training the network. It maps the set of input data to a suitable output set inspired by the way biological nervous systems of the brain process information. Figure 2.13 is an example of MLP operations.

### 2.5.3.6 Logistic Regression (LR)

LR classifier [68] is a mathematical modelling approach used to describe the relationship of a dependent variable (i.e., outcome) and one or more independent variables (i.e., predictors). LR is applicable when the outcome is a binary variable that contains data coded as 1 (yes, success) or 0 (no, failure). Thus, the LR model predicts  $P(Y=1)$  as a function of  $X$ , as illustrated in Figure 2.14.

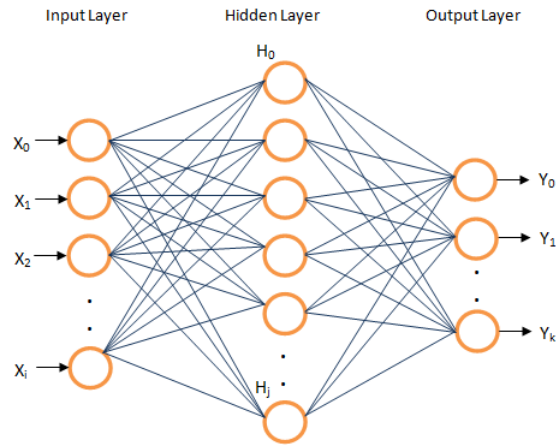


Figure 2.13: MLP technique.

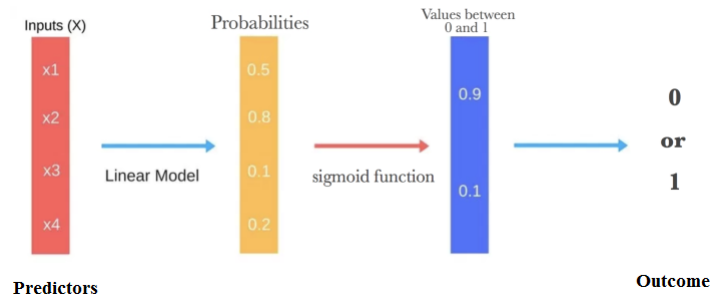


Figure 2.14: Logistic regression technique.

### 2.5.3.7 Deep learning Classifier (DL)

DL is a particular ML technique that implements the learning process elaborating the data through Artificial Neural Networks (ANNs) [62][69]. ANNs have self-learning capabilities that enable them to produce better results as more data becomes available. An ANN has artificial neurons interconnected through at least three layers: the input layer, one or many hidden layers, and the output layer. Each neuron has inputs (e.g., features from a dataset or outputs from other nodes) and produces a single output which can be sent to multiple other neurons. The outputs of the neurons in the output layer return the final result, such as the classification of a sample as an attack or not. Each connection in the network is assigned a weight that represents its relative importance. The weights are adjusted during training to find patterns and make better predictions. Several hyper-parameters need to be set before the learning process begins, such as the number of hidden layers and the number of neurons per layer, the activation function, the learning rate, batch size, and the number of epochs. ANNs are categorised into supervised (e.g., MLP) and unsupervised learning (e.g., DL) [69].

The adjective "deep" in DL comes from the fact that the classification is conducted by training data, with many layers in hierarchical networks with unsupervised

learning. Figure 2.15 depicts the difference between a shallow model (e.g., MLP) and a DL model.

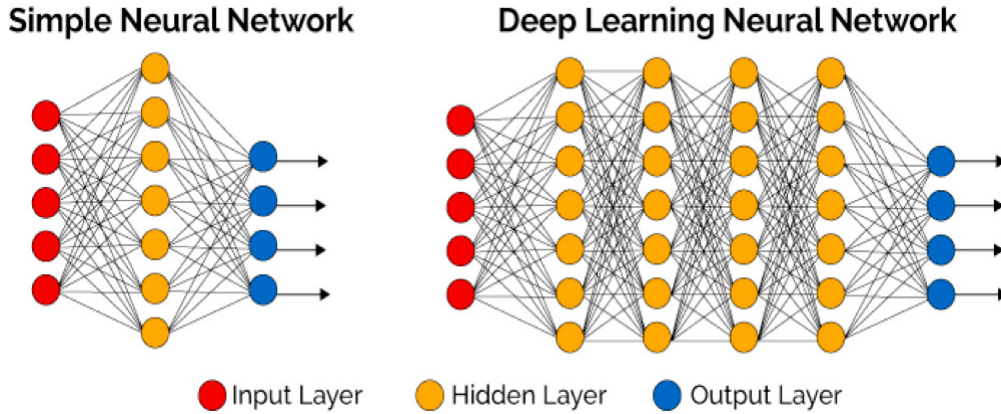


Figure 2.15: Shallow vs DL.

There are several DL models based on the used architectures and techniques [70]. The Deep Recurrent Neural Network (RNN), the Deep Auto-Encoder, the Deep Boltzmann Machine (DBM), and the Deep Believe networks (DBN) belong to the Generative Architecture (GA) class, whereas, the Deep Convolutional Neural Network and the Deep Recurrent Neural Network (when the output is taken to be the predicted input data in the future) belong to the Discriminative Architecture (DA) class. GA models are graphical models that *”are intended to characterise the high-order correlation properties of the observed or visible data for pattern analysis or synthesis purposes, and/or characterise the joint statistical distributions of the visible data and their associated classes.”* [70]. DA models *”are intended to directly provide discriminative power for pattern classification, often by characterising the posterior distributions of classes conditioned on the visible data.”* [70].

## 2.6 Summary

In this chapter, we gave definitions, characteristics, and applications of IoT. Moreover, we addressed the most leading IoT security challenges and issues to be resolved. Besides, we provided an overview of IDS focusing on its definition, types and methods. Finally, we provided an overview of the ML concept, its common terminology and tasks, and we concluded by presenting the different ML algorithms experimented for our IDS.

In the next chapter, we will focus on the RPL protocol and its security issues. We will conclude by giving a synthesis of the existing research works on IDSs for RPL security.

## Chapter 3

# The IPv6 Routing Protocol for LLNs (RPL): Overview, Security Issues, and State-of-the-art Solutions

RPL is a proactive distance-vector IPv6-based routing protocol designed and standardised by the IETF Routing Over Low-power and Lossy networks (ROLL) working group to overcome the routing challenges underpinning LLNs-IoT networks. The RPL specification considers limitations in both the energy power and the computational capabilities of such networks [22]. This chapter provides an overview of RPL's operations (i.e., control messages, upward and downward routes construction, the objective functions to optimise and select routes, and the mechanism for routing maintenance). Besides, it analyses the RPL's security vulnerabilities and presents the state-of-the-art works that address the lack of RPL's security.

### 3.1 RPL Topology Construction

RPL organises the physical network into a logical representation as a Directed Acyclic Graph (DAG) to route traffic/packets. The DAG comprises one or multiple DODAGs (Destination Oriented DAGs) with one root per DODAG. Each root, called border router (BR), is connected to the Internet and other potential BRs via a backbone. Each device/node in the DODAG has many attributes such as an IPv6 address (ID), a list of parents with one preferred parent, a list of discovered neighbours, and a Rank. The Rank of a node identifies the node's position relative to the BR. In the RPL topology, two conditions have to be checked from nodes along the route. Firstly, the Rank values should increase from the BR towards the leaf



nodes, and decrease from the leaf nodes toward the BR. Secondly, packets should be transmitted upward towards the BR, or downward towards leaf nodes, respecting the Rank rule defined in [22]. In other words, when a node receives a packet upward, the sender must have a Rank higher than that node and vice versa, when a node receives a packet downward, the sender must have a Rank lower than that node.

An LLN may run multiple logically independent instances of RPL concurrently. Each such instance may serve different and potentially antagonistic constraints and Objective Functions (OF). An RPL node may belong to various RPL Instances, and it may act as a router in some and as a leaf in others. An RPL Instance is a set of one or more DODAGs that share the same RPL Instance Identifier (RPLInstanceID). DODAGs with the same RPLInstanceID share the same OF (the details of OF are presented in Section 3.2).

RPL supports three communication patterns; the point-to-point (P2P) pattern in which every node may communicate with other nodes in the network, the multipoint-to-point (MP2P) pattern in which data is gathered by a group of nodes and transmitted to one destination that is the BR, and the point-to-multipoint (P2MP) pattern in which the BR sends data to the in-network nodes. MP2P traffic is carried through upward routes, whereas P2MP traffic is carried through downward routes. Figure 3.1 shows a typical RPL topology using two RPL instances running their respective objective functions (i.e., OF1 and OF2) and three DODAGs; two in the first instance and one in the second instance. The figure highlights the different control messages and communication patterns that will be detailed in the next subsections.

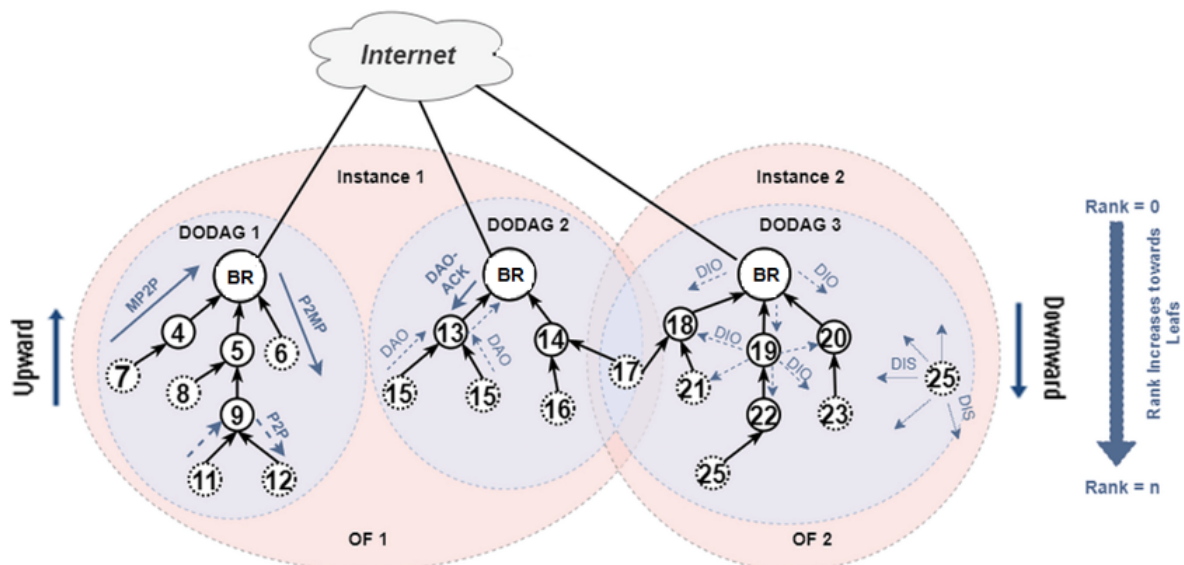


Figure 3.1: The routing protocol for low power and lossy networks.

### 3.1.1 RPL Control Messages

RPL uses the following specific Internet Control Message Protocol (ICMPv6) control messages to exchange routing information, construct and maintain the DODAG.

- The DODAG Information Object (DIO) message conveys the relevant information and configuration parameters that enable a node to join a DODAG, select a set of candidate parents, construct and maintain the DODAG. Among others, the DIO message conveys node and link metrics and constraints (e.g., node energy, hop count, throughput, latency, link colour, and ETX; Expected Transmission Count) [71]. Besides, DIOs carry the OF that the nodes should use to optimise the path construction and calculate their Rank values [72] [73]. Figure 3.2 portrays the DIO message format.

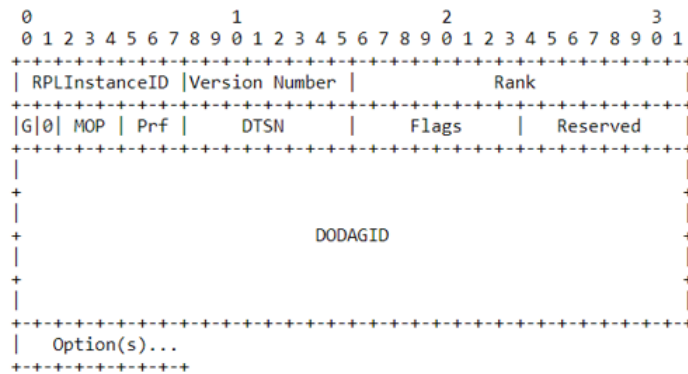


Figure 3.2: DIO format.

- The DODAG Destination Advertisement Object (DAO) messages allow nodes to propagate their destination information upward along the DODAG to the BR using the end-to-end approach. Consequently, the downward routes from the BR to its associated nodes (MP2P) can be constructed and updated (i.e., routing tables' construction and update). Figure 3.3 illustrates a DAO message format.

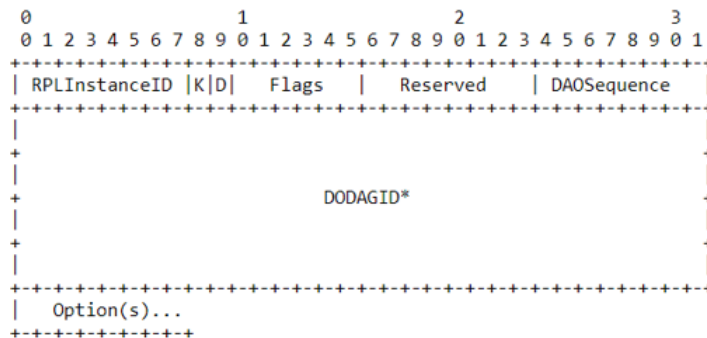


Figure 3.3: DAO format.

- The Destination Advertisement Object Acknowledgement (DAO-ACK) may be unicast by a node to the DAO sender to acknowledge that DAO's reception.
- The DODAG Information Solicitation (DIS) messages aim to discover the neighbourhood and network topology. Precisely, nodes seeking to join a DODAG use DIS messages to solicit a DIO from their neighbours. Figure 3.4 depicts a DIS message format.

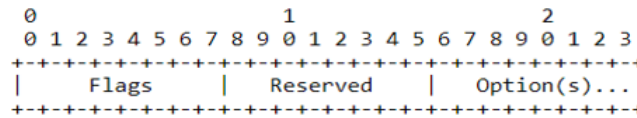


Figure 3.4: DIS format.

### 3.1.2 Upward routes

The DIO messages control the construction and maintenance of the RPL DODAG and upward routes that support MP2P traffic [22]. Figure 3.5 portrays an example of the process to build the DODAG. (a) The BR multicast an initial DIO message announcing configurations (see Figure 3.2), such as its Rank ( $R=0$ ), the RPL InstanceID, the DODAG ID, the DODAG Version, Trickle timer variables, the mode of operation (see Section 3.1.3), the OF, and the metrics/constraints that should be used. (b) When a node receives a DIO message from the BR, it selects the BR as its parent, calculates its Rank ( $R=1$ ), sends a DAO to its parent, and broadcasts an updated DIO to its neighbours. (c) On receiving DIOs from nodes of Rank 1, each neighbour adds the sender address to its candidate parents set, selects a preferred parent of Rank 1, calculates its Rank ( $R=2$ ), sends a DAO to its parent (as in Section 3.3), and multicast an updated DIO with its own Rank to its neighbours. The nodes may discard a received DIO as specified in Section 3.3. (d) All neighbouring nodes repeat the process until each node joins the RPL network. Thus, when a node needs to send a packet to the BR, it will forward the packet to its own parent, which in turns will forward it upward until it reaches the BR. Figure 3.6a is an example of a simple network before routes construction, while Figure 3.6b illustrates the data packets forwarding following the constructed upward routes. After the construction of the RPL DODAG, the maintenance begins respecting the Trickle timer defined in Section 3.3.

### 3.1.3 Downward routes

Downward routes are constructed to support P2MP and P2P traffic flows. RPL uses DAO messages that are propagated upwards in the DODAG topology via a parent

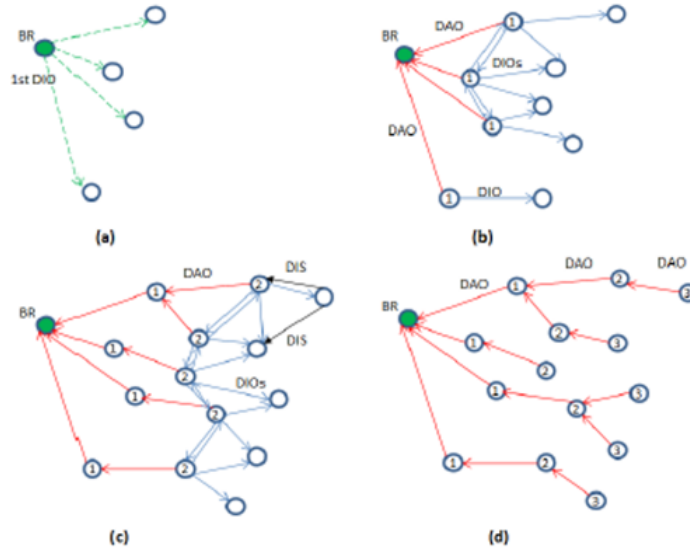


Figure 3.5: RPL DODAG and upward routes construction.

to construct and maintain downward routes. A node that wants to be reachable by the BR unicasts a DAO to its preferred parent with its own destination prefix [22]. The received DAO by the parent will be processed according to one of two modes of operation: storing and non-storing.

### 3.1.3.1 Storing Mode

In the storing mode, each node keeps a routing table of all destinations reachable via its sub-DODAG and their respective next-hop nodes from received DAOs. Indeed, when a parent receives a DAO from one of its children, it stores the announced destination in its routing table along with the DAO sender address as the next hop to reach that destination. Next, the parent forwards the received DAO to its preferred parent to ensure the advertised destination's propagation upward to the BR. Each intermediate node repeats the process until the BR finally receives the DAO [22][74]. In this mode, data packets are forwarded upwards until they reach a node with routing information about the destination. Once a common ancestor is reached, the packets proceed downwards following the routes previously established by the DAO messages, as illustrated in Figure 3.6c and Figure 3.6d.

### 3.1.3.2 Non-Storing Mode

In the non-storing mode, a parent receiving a DAO does not store any routing state. The BR is the only node maintaining routing information. It exploits the information in DAOs for source routing (i.e., the BR includes routing information directly into the packet itself). The intermediate nodes (parents) simply forward the received DAO messages to their respective preferred parents until the BR finally

receives the DAOs [22][74]. In this mode of operation, data packets must travel upwards all the way to the BR as it is the only node maintaining routing information before being redirected to their destination (Figure 3.6e).

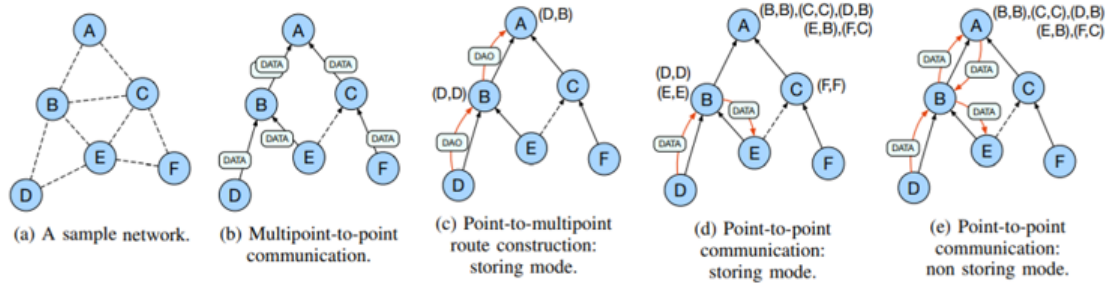


Figure 3.6: Routing in RPL. Existing routes are shown next to the network nodes [5].

## 3.2 Objective Functions

The Objective Function (OF) is a route selection and optimization mechanism towards the root node (BR). Principally, OF is made of a set of rules that combine the nodes and links metrics and constraints to calculate the Rank and define how this latter should be used for selecting the preferred parent. RPL may use different OF to meet different applications' requirements, such as minimizing the energy consumption or maximizing reliability. There exist two standardised OFs, which are the Objective Function Zero (OF0) [72] and the Minimum Rank with Hysteresis Objective Function (MRHOF) [73].

### 3.2.1 The Objective Function Zero (OF0)

According to OF0, each node selects as the preferred parent the node with the minimum value of hops to reach the DODAG root [72]. Furthermore, each node calculates its Rank ( $R_n$ ) as the sum of its selected preferred parent's Rank ( $R_p$ ) and a strictly positive scalar value (rank\_increase) as depicted in Equation 3.1 and Equation 3.2. In Equation 3.2,  $S_p$  is the step-of-Rank representing a value related to the parent link metric and other properties such as the hop-count.  $R_f$  and  $S_r$  are normalisation factors called the Rank factor and the stretch of Rank, respectively. MinHopRankIncrease is the minimum increase in Rank between a node and any of its DODAG parents [72]. The OF0 does not specify which metric should be involved in the calculation of Rank increase. In OF0, each node considers the parent with the least possible Rank as its preferred parent for parent selection. It also selects

another parent as a backup if the connectivity with its preferred parent is lost [72].

$$R_n = R_n + \text{rank\_increase} \quad (3.1)$$

$$\text{rank\_increase} = (R_f * S_p + S_r) + \text{MinHopRankIncrease} \quad (3.2)$$

### 3.2.2 Minimum Rank with Hysteresis Objective Function (MRHOF)

In MRHOF, each node selects the route that minimises a metric, while using hysteresis to reduce the frequent change of the preferred parent in response to small metric changes [73]. The metrics to use are advertised in the DAG Metric Container in the DIO messages. The standard MRHOF uses the energy constraint or the Expected Transmission Count (ETX) metric to calculate the node's Rank and select the preferred parent in the DODAG formation. ETX represents the number of expected transmissions and retransmission of a packet necessary for it to be received without error at its destination. ETX aims to select routes with high end-to-end throughput and is defined as in Equation 3.3, where  $d_f$  is the forward delivery ratio (i.e., the probability that a packet is received by a neighbour) and  $d_r$  is the reverse delivery ratio (i.e., the probability that an acknowledgement packet is successfully received).

$$\text{ETX} = \frac{1}{d_f * d_r} \quad (3.3)$$

After calculating the path costs through all candidate parents, a node selects the parent with the lowest path cost as its preferred parent. Nonetheless, MRHOF allows a node to change its parent only when the new path differs from the old one by at least PARENT\_SWITCH\_THRESHOLD, which is the hysteresis part of MRHOF. MRHOF hysteresis yields a trade-off between route stability and optimality. Thus, if the threshold is too high, parents are less likely to change and routes are more stable; nonetheless, their quality may degrade significantly before they are reconfigured [73].

### 3.2.3 Routing Metrics for Path Calculation

OF needs to use one or multiple metrics and constraints to determine the best path and calculate Rank. The RFC 6551 [71] specifies a set of link and node routing metrics and constraints that can be used by RPL to meet LLNs applications' requirements. The metrics and constraints are carried within the DAG Metric Container object (MC) of the DIO [22]. Besides, they can be used separately or combined within the OF. An example of using separated metrics, an OF can use the aggregation of the remaining node energy metric along the path by applying a Min function

to select the minimum energy value at each hop. Afterwards, the OF use a Max function that compares the energy from several paths and selects the best path as the one with the maximum energy value.

Another example is with an OF that combines a link metric (ETX) and a node constraint (Energy). In the case of a DODAG where all nodes must be mains-powered and the best path is the one with lower aggregated ETX, the MC will carry two routing objects; one is an ETX metric object and the second one is a Node Energy constraint object. Indeed, an RPL instance may use the metric object to report a maximum, a minimum or the aggregation. Consequently, if the best path is the one avoiding low-quality links, then the path metric reports a maximum (i.e., the higher the ETX, the lower the link quality). Thus, when a node processes the DIO message reporting the link metric (ETX), each node selecting the advertising node as a parent updates the value carried in the metric object by replacing it with its local link ETX value if and only if the latter is higher. On the other hand, if the constraint object indicates that nodes must be mains-powered and the constraint signalled in the DIO message is not satisfied, the advertising node is just not selected as a parent by the node that processes the DIO message.

### 3.3 Trickle Timer

RPL uses a Trickle algorithm that regulates DIO control messages' transmission rate according to the current network conditions. Trickle increases the transmission rate when a change in routing information is detected (e.g., altered DIO messages, a node joining the DODAG, etc.) to update the network rapidly with new information. In a steady case, Trickle exponentially reduces the transmission rate to limit the number of transmissions when there is no update to propagate. On the other hand, Trickle maintains a suppression mechanism in which a node limits redundant messages. Hence, the node suppresses the scheduled control packets if it detects that enough of its neighbours have transmitted the same piece of information [75][6].

The Trickle algorithm involves three configuration parameters: 1) the maximum interval size ( $I_{\max}$ ); 2) the minimum interval size ( $I_{\min}$ ); 3) the redundancy constant ( $k$ ). Furthermore, it maintains three variables: 1) the size of the current interval ( $I$ ); 2) a counter ( $c$ ); 3) a specific time within the current interval ( $t$ ). Each node is responsible for handling its interval. The interval boundaries are  $[I_{\min}, I_{\max}]$ . This interval is divided into sub-intervals (periods). The first sub-interval starts with  $I_{\text{start}} = I_{\min}$  and ends with  $I_{\text{end}} = I_{\text{start}} * 2$ . Each time the first sub-interval is finished, a new sub-interval starts until reaching the end of the primary interval (i.e.,  $I_{\max}$ ), as illustrated in Figure 3.7. In the case of RPL, whenever a node hears a consistent DIO transmission from its neighbours, it increments the counter 'c'. At time  $t$ , the node

transmits Multicast DIO if the counter 'c' is less than the redundancy constant 'k'. If not, the node suppresses the scheduled DIO transmission, waits until the current sub-interval 'I' has expired, and then doubles the sub-interval length. Each time the node needs to check if it reaches the maximum of the interval  $I_{\max}$  [6][75].

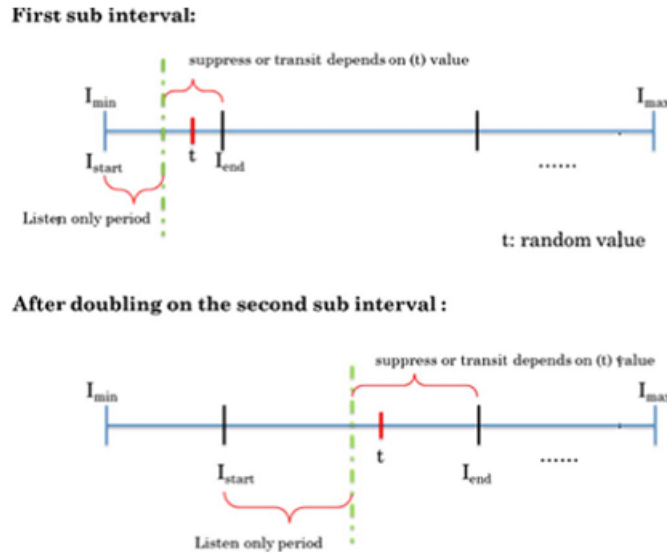


Figure 3.7: The Trickle algorithm for a node [6].

## 3.4 RPL Tools

This section presents the most used open-source OSs, simulator and hardware by the RPL research community in the IoT domain.

### 3.4.1 Operating Systems

#### 3.4.1.1 ContikiOS

ContikiOS is a lightweight and portable open-source operating system designed specifically for low-power resource-constrained IoT objects [76][77]. Contiki has two main partitions the core and the loaded programs, as illustrated in Figure 3.8 [7]. The core contains the kernel, libraries, the program loader, device drivers, and the communication stack for the communication hardware. Two types of event are supported within the Contiki kernel: asynchronous and synchronous events. Thus, Contiki provides the real-time clock for synchronisation purposes and real-time applications. Contiki supports the uIPv6 stack in addition to several other IoT standards such as 6LoWPAN and CoAP. It also implements the RPL standard operations via the ContikiRPL library and both OF0 and MRHOF. However, ContikiRPL does not include any RPL security features [78].



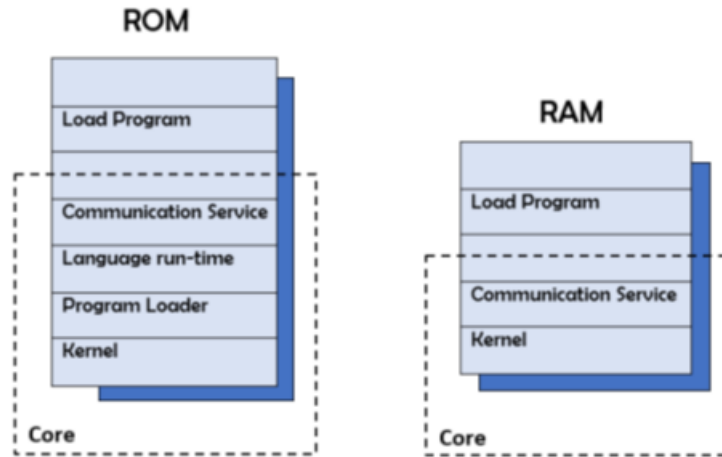


Figure 3.8: ContikiOS’s core and loaded programs [7].

### 3.4.1.2 TinyOS

TinyOS [79] is an open-source operating system designed for WSNs. TinyOS implements TinyRPL, the RPL library that provides all basic features of the RPL standard. It is designed to be used with the Berkeley Low-power IPv6 stack (BLIP) provided in TinyOS. TinyRPL supports both storing and non-storing modes with the default upward routes, in addition to both the OF0 and MRHOF. Nevertheless, similarly to ContikiRPL, TinyRPL lacks any support for RPL security features.

### 3.4.1.3 RIOT

RIOT [80] is an operating system for memory-constrained low-power wireless IoT devices. RIOT has its own implementation of the RPL standard named RIOTRPL [81]. RIOTRPL supports the two downward RPL’s modes (storing and non-storing). Nonetheless, it only implements the OF0 and does not support any security modes.

## 3.4.2 Cooja Simulator

The Cooja simulator [82] is a java-based network cross-level simulator based on Contiki OS and specifically designed for LNNs. Unlike other simulators such as NS2 and OMNET, Cooja is an emulation and simulation tool where the software of the sensor node itself can be run either on the simulator as compiled native code for the platform or on the emulator as an actual sensor node MSP430 at the hardware level. Furthermore, Cooja has the particularity that the same experimenting code running on the emulator will run on real IoT devices. Besides, the simulation can be conducted for various types of motes within a single simulation. Otherwise, as an emulator, Cooja gives a lot of specifics regarding the mote’s hardware.

### 3.4.3 Hardware

Figure 3.9 depicts a list of various types of nodes that are used in IoT application domains. the Tmote Sky also known as TelosB nodes are widely used among the research community of IoT.

Architecture	Model	Radio-Chip	MCU	RAM	Flash
AVR	MicaZ	CC2420	ATmega128L	4 KB	128 KB
	Waspnote	8 radios	ATmega128L	8 KB	128 KB
MSP430	Skymote	CC2420	MSP430F1611	48KB 10KB 1MB	48 KB
	XM1000	CC2420	MSP430F268	8 KB	116 KB
ARM	CC2538	integrated 802.15.4 radio	ARM7MC13224	16 KB 32 KB	128 KB 256 KB 512 KB
	Econotag	integrated 802.15.4 radio	ARM7MC13224	96 KB	128 KB

Figure 3.9: Specifications of typical constrained devices [8].

## 3.5 RPL Security

The current RPL specification includes a few self-healing mechanisms, like loop detection and avoidance, global and local repair mechanisms. Furthermore, it defines security features like cryptographic security modes that are presented in the following subsections.

### 3.5.1 Self-healing and Fault Tolerance Mechanisms

#### 3.5.1.1 Loop Detection and Avoidance

Loops may form for several reasons, such as control messages loss or malicious behaviour. RPL includes a reactive loop detection and avoidance technique that uses RPL Packet Information, which transported with data packets, and the Rank rule presented in Section 3.1 [22]. The RPL Packet Information placed in an IPv6 Hop-by-Hop option-header includes the Rank of the sender field along with the direction of the packet flag (i.e., upward or downward). If a Rank rule inconsistency is detected (e.g., if a node receives a packet flagged as moving in the upward direction, and if that packet carries that the sender is of a lower Rank than the receiver, then the

latter must conclude that it may be a loop.), the receiver node must trigger the local repair mechanism.

### 3.5.1.2 Global and Local Repairs

RPL provides global repair (GR) and local repair (LR) mechanisms to fix links and node failures and detect loops and other inconsistencies. GR is only triggered by the BR (DODAG root). Each time the BR decides to form a new version of the DODAG it initiates a GR by incrementing the DODAG Version Number field within the DIO message. Note that a GR starts the reconstruction of the whole topology from scratch. On the other hand, any non-root node that detects an inconsistency (e.g., loop or link failure) can start an LR. In the LR mechanism, the node should poison its routes by announcing a Rank of INFINITE RANK. Therefore, it detaches itself from the DODAG and then re-attaches to the DODAG as a new joining node using a DIS message [22].

## 3.5.2 Security Features

The self-organising, self-healing, and resource-constrained, as well as unreliable links, limited physical security, and dynamic topology of RPL networks, expose them to various internal and external threats. The RFC 6550 [22] states that RPL could use link-layer security mechanisms when they are available to secure message transmission. Furthermore, the RPL specification defines the following optional cryptographic security modes that nodes within an RPL network can adopt to ensure communication security.

### 3.5.2.1 Unsecure Mode

In this mode, RPL control messages presented in Section 3.1.1 are transmitted without any additional security features [22]. In this case, RPL relies on other layer security primitives, such as the MAC layer, to satisfy the network's security requirements [22].

### 3.5.2.2 Pre-installed Security Mode

In this mode, the nodes use a secured version of control messages. Accordingly, the nodes have pre-installed keys to generate and process RPL secured messages and thus provide control messages confidentiality, integrity, and authenticity [22].

### 3.5.2.3 Authenticated Security Mode

Like the pre-installed mode, the nodes use a secured version of control messages and have pre-installed keys. Nevertheless, they may only use the keys to join the network as a leaf. A router that needs to enter an RPL network requires another key from an authentication authority [22].

## 3.6 RPL Vulnerabilities

### 3.6.1 RPL Security Limitations

As presented in the sections above, the RPL specification defines some fault tolerance mechanisms and secure versions of the different RPL control messages (i.e., control messages encryption). The defined security mechanisms ensure control messages integrity and confidentiality against outsider attackers [22]. However, on the one hand, the proposed security mechanisms are not clearly defined; for instance, it is not specified how asymmetric cryptography may be employed to support node authentication and key retrieval. On the other hand, the objects are not tamper-resistant, which means that an adversary can compromise some of them, reprogram and redeploy them into the network as insider attackers. The insider attacker can bypass the defined security mechanisms by gaining access to shared keys and trigger several attacks against RPL to break routing operation rules and cause network disruption for different IoT applications. Actually, attacker nodes can exploit RPL's mechanisms to compromise the legitimate nodes. Malicious nodes may falsify or lie the advertisement of link and node routing metrics to disturb traffic routing. Besides, they may exploit both global and local repairs to trigger specific attacks against RPL networks (see Section 3.6.2). Accordingly, in RPL, each node selects only one preferred parent through which all traffic will be forwarded, as long as it is reachable. An adversary can easily identify the preferred parent, target it and trigger different attacks. Furthermore, RPL have two further gaps that are the lack of identification mechanisms and the lack of supporting mobility. For instance, the fact that RPL uses IPv6 addresses as nodes' identifiers makes the routing protocol vulnerable to Sybil attacks, in which an adversary creates easily fake IPv6 addresses to participate in the network operations as legitimate nodes. Furthermore, because RPL does not handle mobility when routing, malicious nodes could trigger mobility-based attacks. Section 3.6.2 gives details on how an adversary can exploit the limitations of RPL to trigger attacks and harm the network.

### 3.6.2 Attacks against RPL

Attacks against RPL networks can be passive or active. Passive attackers eavesdrop on communications to intercept data packets or information on the network, whilst active attackers exploit RPL weaknesses to break into the network in order to introduce, alter, or delete data, and could also destroy part or even the whole network. Except for the eavesdropping attack, this section classifies attacks against RPL as active since they aim to disrupt and/or destroy the network.

Several classifications for RPL threats exist in the literature. Pongle and Chavan put all attacks in the same category, namely attacks on RPL topology [25]. The authors in [26] have classified RPL attacks into three categories; attacks on resources, attacks on topology and attacks on traffic. Based on the resource-constrained nature of IoT nodes, we consider that both attacks on topology and traffic affect directly or indirectly the network resources, which means that all attacks can be classified as attacks on resources. Tsao et al. proposed four categories of RPL attacks under the ISO 7498-2 model; attacks due to failures to authenticate, attacks due to failures to keep routing information confidential, attacks on integrity, and attacks on availability [24]. Like in the work of Tsao et al. [24], Airehrour et al. presented a summary of attacks against RPL and classified these attacks as confidentiality, integrity and/or availability attacks [83].

From our point of view, attacks against RPL can be classified into two main classes: the Novel RPL Specification-based attacks, such as Rank, neighbour, and version number attacks, and the Existing routing attacks tailored to the context of RPL, such as spoofing, hello flood, homing, selective forwarding, Sybil, wormhole, acknowledgement flooding, eavesdropping, impersonation, relay, and replay attacks. In the following, we present an exhaustive study of existing attacks against RPL.

#### 3.6.2.1 Novel RPL Specification-Based Attacks

This class includes new threats exploiting some operating rules in the RPL specification. This kind of attacks may cause control messages overhead, discard downward routing state, and exhaust nodes resources.

**3.6.2.1.1 Rank attacks** In the literature, there are several variants of the Rank attack. These variants are termed Rank attacks, Increased Rank attacks, Decreased Rank Attacks, and Worst Parent Attacks. Every one of these variants can be simply called Rank attack because they are based on the malicious manipulation and/or exploitation of the Rank field and/or rules.

- In [84] and [85] Rank attack, two or more malicious nodes may misbehave and cooperate on skipping the Rank checking function. This leads to break

the Rank rule and create un-optimised routes, or undetectable loops, or even never discovering existing optimised paths.

- In [86] Rank attack, the attacker chooses a random parent with a Rank higher than the preferred parent's Rank (worst parent attack). This attack creates un-optimised routes, which leads to poor performance.
- In the Rank attack defined in [87], [88] and [89], an adversary node illegitimately advertises a better Rank equal to a lower Rank value (decreased Rank attack) to attract the traffic.
- Rehman et al. [90] introduced a new variant of this attack termed the Rank Attack using Objective Function (RAOF attack), where the advertised Rank value is bounded. In this variant, the attacker announces a Rank value less than its neighbours' minimum Rank and greater than its preferred parent. In addition, because RPL suggests no measures to monitor the change in routing metric values, the attacker announces in the DIO message a drastically lowered value of the routing metric compared to the minimum observed among its neighbours. In both attacks, the honest neighbours will select the attacker as their new preferred parent, and thus allowing it to manage and manipulate more network traffic. Consequently, they enable the malicious node to trigger other attacks (e.g., eavesdropping, deleting and modifying data).
- In [26] Rank attack, the attacker voluntarily increases its Rank value (i.e., increased Rank attack), which leads to generate loops in the network, exhaust node resources, and congest the network.

**3.6.2.1.2 Neighbour attack** In the Neighbour attack, misbehaving nodes re-send replicated DIO messages (without updates) to honest nodes [86]. Hence, honest nodes can consider the owners of the message as neighbours. Consequently, they can update their routes to the out range neighbours, which leads to creating false routes, disrupting the network, or consuming more resources. As described, the Neighbour attack is more like a Routing information replay attack where the attacker forwards outdated DIO messages inducing honest nodes to update their routing tables with stale routes [26]. This attack has also been termed DIO replay attack.

**3.6.2.1.3 Routing Choice Intrusion attack** Nodes use DIO messages metrics and objective function rules to decide whether to join the DODAG or not. The authors in [91] defined a new RPL internal attack where the attacker learns RPL's routing rules (i.e., choices to route packets), captures control messages and broadcasts fake ones. According to the authors, this attack is hard to detect because

the intruder has only to ignore the legitimate RPL internal detection and work as normal. One or more attackers may participate in the attack. This attack can generate un-optimised routes, create loops, and exhaust resources. In a variant of this attack, attackers tamper the DIO metric value and ignore inside legitimate metric detection. This variant is identical to the RAOF attack.

**3.6.2.1.4 DAO attacks** The DAO messages are used to build and maintain the downward routes to carry the traffic from the BR to the respective nodes. Thus, after a node joins a DODAG, it advertises a DAO message for its neighbours to update their routing tables. Besides, parents can use DIO messages to request DAO from sub-DODAG. Attackers can exploit the DAO mechanisms to trigger several variants of DAO attacks as follows.

- In the DAO-Inconsistency attack, the attackers can use the Forwarding-Error F flag to make a node discard available downward routes [92]. This attack aims to make the DODAG's topology sub-optimal and to isolate the sub-DODAG bound the attacker.
- In the Routing Table Overload and the Routing Table Falsification attacks, the malicious nodes announce fake routes by modifying or forging DAO messages [26], leading to build false downward routes and overload the targeted nodes' routing tables with these false paths. Consequently, honest nodes will be prevented from building new legitimate routes. Additionally, used paths can be longer, resulting in delay, packet drops and/or network congestion.
- In the DAO Insider attack [93][94], the attackers repeatedly replay eavesdropped DAO messages from legitimate nodes. This attack aims to drain the network resources. The authors proposed SecRPL, a solution to address the attack. SecRPL restricts the number of forwarded DAOs by a parent per destination.
- In the DAO Induction attack [95], the attackers repeatedly send DAO messages while incrementing the DAO-DTSN field. In fact, the number of time an attacker can increment DTSN is unlimited.

**3.6.2.1.5 DIS attacks** RPL is based on the IPv6 Neighbour Discovery mechanism. It relies on Multicast operations to set up the network topology. A node within an RPL network sends a DIS message to solicit DIO messages from neighbouring nodes and join the DODAG. The DIS transmission interval varies from one RPL's implementation to another. For instance, in the RPL Cooja-Contiki simulator [82], it is handled using `RPL_CONF_DIS_START_DELAY` and `RPL_CONF_DIS_INTERVAL`

constants. After booting, a node delays the transmission of its first DIS message according to the `RPL_CONF_DIS_START_DELAY` value. A node aiming to join the network continuously transmits DIS messages within the `RPL_CONF_DIS_INTERVAL` fixed interval until it receives a DIO message from its neighbours. Upon receiving a DIO message, it stops transmitting DIS messages and joins the network by sending a DAO message to its selected parent. The DIS message transmission can be unicast or multicast. A malicious node can exploit the DIS mechanisms to trigger the next attacks.

- In the DIS Multicast attack, the malicious node multicasts periodically or continuously DIS messages to its neighbours. On receiving a DIS multicast, the neighbouring nodes have to reset their Trickle timers and transmit multicast DIO messages more frequently. The repeated forced rest of the Trickle timer leads to increase the number of control messages sent by the nodes, flood the network with fake messages, disrupt the network operation, exhaust resources, and further reduce the network lifetime [86][96][97][98]. These attacks were defined in [26] as flooding attacks.
- In the DIS Unicast attack, the malicious node sends periodically or continuously unicast fake DIS messages to the nodes in its neighbour list or a target node. On receiving a unicast DIS message, the normal node replies to the sender with a unicast DIO message without resetting its Trickle timer. This attack affects the control packet overhead and overall power consumption of the network. Nevertheless, the DIS Unicast attack's impact on the network is less compared to the DIS Multicast attack.

The authors in [96] conducted extensive simulation experiments to evaluate the performance of RPL under the Spam DIS attack. According to their results, the attack significantly increases energy consumption and decreases the node lifetime. They concluded that the DIS attack is an extremely severe Denial of Service (DoS) attack for RPL-based LLNs. It has been demonstrated that the DIS attack negatively impacts the usage of nodes' resources with a decrease of 2% in LPM and an increase of 226%, 1275%, 81%, and 171% in the CPU Time, TX (transmitting) Time, RX (receiving) Time, and battery consumption, respectively [99]. In another work [100], the DIS attack's effects on energy efficiency and the DODAG construction have been examined. The simulations results demonstrated that the malicious node's neighbours are highly affected by the attack in terms of power consumption, then the nodes present at extreme boundaries. Indeed, the interference increases for all nodes with the presence of a malicious node. Accordingly, the ON and transmission periods increase, especially for the neighbours of the malicious node. Furthermore,



the attack affects the DODAG construction in the malicious node's transmission range.

**3.6.2.1.6 Version number attack** The DIO message version number field is set and updated only by the BR. Each time a rebuilding of the DODAG is necessary, the DODAG version number is incremented by the BR and propagated unchanged down the DODAG graph. This process is known as global repair. As there is no mechanism in RPL to protect the version number field from modifications, an adversary can illegitimately increase the version number of the DODAG, which triggers the global repair mechanism and thus the reconstruction of the RPL topology from scratch. This attack leads to increase the control messages overhead, generate loops and un-optimised topology, and hence exhaust resources. According to the state-of-the-art, the version number attack is more effective when the attacker is located far from the BR [87][101][102].

**3.6.2.1.7 Local repair attack** An attacker can repeatedly trigger a local repair by changing the DODAG ID field or broadcasting infinite Rank, which leads to update the network topology, and thus consume more resources [72][84][85]. Furthermore, to trigger a local repair attack, compromised nodes can modify the Down 'O' flag and Sender-Rank field. Indeed, this attack may be triggered just by modifying flags or adding new flags in the header. The latter attack is defined in [24] as the DAG inconsistency attack. As defined, the local repair attack is more like a poisoning or detaching attack.

**3.6.2.1.8 Resource depleting attacks** Le et al. defined the resource depleting attacks as being the ones triggered when an adversary initiates greedy activities aiming to exhaust the nodes' resources [84][85]. From our point of view, all attacks against RPL affect directly or indirectly nodes resources. The fact that attacks create un-optimised routes, generate loops or congest the network, all these factors lead honest nodes to consume more resources.

### 3.6.2.2 Existing Attacks Tailored to the Context of RPL

This class includes well-known routing attacks, which have already been studied by the research communities and have been tailored to the context of RPL.

**3.6.2.2.1 HELLO Flood Attacks** In the literature, there exist several forms of the hello flooding attack against RPL.

- In an RPL network, an attacker can introduce itself as a neighbour to nodes within the network by broadcasting DIO messages -as a HELLO message- with

strong signal power and a favourable routing metric. If nodes send packets to the attacker, their messages may get lost because the attacker might be out of range [103]. From our point of view, this attack looks like the neighbour attack.

- As already said, nodes aiming to join an RPL topology could send DIS messages to solicit DIOs from their neighbours. In a second form of the hello flooding attack, a malicious RPL node could flood the network with a massive amount of DIS messages, causing the recipient nodes to respond by sending DIO messages [97][104].

**3.6.2.2.2 Sinkhole attack** In this attack, the malicious node advertises itself as the best path with the aim to be chosen as a preferred parent by its neighbours and thus route traffic through it. In our opinion, this attack is similar to the Rank attack, where a malicious node advertises an artificial beneficial Rank to be selected as a preferred parent. As it is, this attack does not appear to be harmful (i.e., passive attack). However, it becomes harmful (i.e., active attack) if combined with other attacks [105].

**3.6.2.2.3 Black-hole attacks** An intruder triggers a black-hole attack by dropping all data packets routed through it. This attack can be considered a DoS attack. Indeed, the black-hole attack is more dangerous if combined with the Rank or sinkhole attacks since the attacker is in a position where huge traffic is routed through it. This attack increases the number of exchanged DIO messages leading to the network's instability, data packets delay and thus resources exhausting [105][106][107][108].

**3.6.2.2.4 Selective-forwarding attacks** In the selective-forwarding attacks, a misbehaving node can either aggressively filter RPL control messages or drop data packets and forward only the control messages traffic. The first attack affects negatively the topology construction and the network functions, which leads to disrupt routing. In comparison, the second attack leads to a DoS attack because no data will be transmitted to destination nodes. These attacks are also known as grey-hole attacks that are a special case of black-hole attack. These attacks are more dangerous and cause great harm if combined with other attacks such as the sinkhole or Rank attacks [106][58][87][103].

**3.6.2.2.5 Wormhole attack** Two or multiple attackers have to connect via wired or wireless links called tunnels to trigger a wormhole attack. A wormhole

attack permits an attacker to replay the network traffic in the other ends of the tunnels. In RPL, some attackers can be outside the 6LoWPAN and thus can bypass the BR. If control messages are replayed to another part of the network, distant nodes see each other as if they are neighbours, which leads to distort routing paths and create un-optimised routes [26]. The wormhole attack is highly harmful especially if combined with other attacks [103].

**3.6.2.2.6 Sybil and CloneID attacks** Sybil and CloneID attacks are similar and known as identity attacks [24]. In the CloneID attack (or spoofing attack), an attacker copies the same logical identity on several physical nodes. In a Sybil attack (or impersonation attack), an attacker copies several logical identities on one physical node. A malicious node can trigger these attacks to access the traffic, take control of the network, or overcome a voting scheme [58][103]. The Sybil attack can be combined with other attacks to affect the network operations harmfully. In the next chapter we introduce and analyse a new attack, named SybM attack. In SybM, the attackers use periodically new fabricated identities (i.e., Sybil identities) and trigger a Sybil-mobile attack to overload the network with fake messages and thus exhaust nodes' resources [28][29].

**3.6.2.2.7 Denial of Service attacks** DoS and DDoS attacks aim to make nodes and/or the network unavailable. These attacks can be triggered against any layer of the IoT architecture. These attacks are simple to implement and very common because they have devastating consequences on the network [57]. From our perspective, the attacks mentioned above may be categorised as DoS or DDoS attacks since they overload the network with fake messages and exhaust resource, which makes parts of the network isolated and unavailable.

**3.6.2.2.8 Indirect attacks** Flooding, jamming and overwhelming are attacks that indirectly affect RPL routing operations, and perform DoS attacks against the network. Indeed, these attacks downgrade the node operation by resource consuming or destroy the network traffic. Flooding and overwhelming attacks are initiated by sending a large amount of traffic to a specific destination network to consume devices' resources. A jamming attack is triggered when an attacker exploits the transmission of a radio signal to interfere with radio frequencies being used by the network. It is initiated by sending forged packets to create collisions; thereby, dropping legitimate packets. Other indirect passive attacks are eavesdropping (i.e., sniffing) and traffic analysis attacks. In both attacks, attackers listen to the packets transmitted over the network. These packets could be data packets, routing data (i.e., DIO, DAO, etc.), and/or partial topology (i.e., parent-child relations). By analysing the gathered

information, attackers may trigger more harmful attacks [24][26][109][110][111].

## 3.7 Security Enhancements for RPL

### 3.7.1 Cryptography-based Solutions

The RPL specification introduces data confidentiality using Message Integrity Code, data authenticity using encryption, and replay protection using the Consistency Check (CC) message [72] for DIO, DAO and DIS control messages. Perazzo et al. [112] implemented and evaluated the different RPL security modes; the unsecured mode, the pre-installed mode with light-security configuration, and the pre-installed mode with full-security configuration. The authors reported that the network formation time, the overhead introduced by the replay protection, and the power consumption introduced by the security features increase with the network size. However, globally the RPL security mechanisms have a negligible impact on the performances if there is no replay attack. Otherwise, if there is a need to protect the network against replay attacks, the impact on performances is more pronounced.

As pointed out in [53] and [50], node authentication can solve most of the problems that may be caused by unauthorised use such as Sybil and CloneID attacks. However, securely managing, processing, and storing cryptographic keys inside a resource-constrained and tamper-resistant embedded device deployed in an unstructured, distributed and untrusted environment is a challenging problem. In this context, several works focused on secure key management mechanisms for IoT. For instance, the authors in [55] proposed Internet Key Exchange (IKE) compression scheme to provide a lightweight automatic way to establish security associations for IPsec. Abdmeziem et al. [113] proposed a compression scheme for the MIKEY-Ticket key exchange protocol to provide a lightweight and energy-aware version for e-health application use.

VeRA (Version Number and Rank Authentication) [87] is a security scheme that has been proposed to provide defence against version number and rank change attacks. VeRA proposes to use hash chains for authenticating the nodes whose rank or version number is changed. The main drawback of VeRA is that it can be bypassed using rank forgery and replay attacks.

### 3.7.2 Trust-based Solutions

The existing cryptographic mechanisms will fail in safeguarding all network aspects since several RPL-based attacks, such as selective-forwarding and black-hole attacks could not be prevented. In this context, tamper-resistant modules and trusted computing technologies are required. A growing number of works exist to secure

RPL based on trust computation. For instance, authors in [114][115][116] introduced a new trust-based metric to use when constructing the RPL topology. In this trust-based RPL, nodes cooperate to calculate the trust metric of their respective neighbours based on nodes behaviours and some trust components (e.g., energy, ETX, and honesty). If a node is detected as untrusted, it will be discarded from the list of parent and a local repair is triggered. Airehrour et al. [83][108] proposed SecTrust-RPL: a trust-aware RPL routing protocol to secure RPL from routing attacks. In SecTrust, the trustworthiness of a node is calculated relying on direct and indirect packet forwarding behaviour between linked and 2-hops nodes, respectively. Although SecTrust uses indirect trust observation, a node recommendation depends only on the neighbour of its indirectly linked neighbours (i.e., the parent of its parent). In other words, the indirect trust of a node is calculated based only on one recommendation of the intermediate neighbour, which makes it vulnerable to Bad-mouthing and Good-mouthing attacks. Khan et al. [117] proposed a centralised trust-based model for managing the reputation of every node participating in RPL-based networks. Every node relies on packets routed across the network to calculate direct trust for other nodes, thus elaborating positive and negative experiences with other nodes. The gathered trust information is then transmitted to a central entity, which evaluates the interactions between network nodes and gives them a global reputation. This solution is vulnerable to a single point of failure.

### 3.7.3 IDS-based Solutions

The mechanisms above can be considered as the first line for solving some RPL security issues. However, they need to cooperate with Intrusion Detection Systems (IDSs), which can monitor and detect malicious nodes from the early phase to eliminate further damage of the attacks. Some pieces of information can be monitored and used to mitigate and/or minimise some attacks impacts on RPL-based networks [72]. For instance, the following numbers can be bounded within a given time in such a way the attacks cannot be triggered several times and to quarantine neighbours having suspicious activities at unacceptable rates.

- The number of times a local repair procedure was triggered (Local repair attack).
- The number of times a global repair was triggered by the BR (Version number attack).
- The number of received malformed messages (Local repair attack).
- The number of times a node request to join a DODAG (DIS attacks).

- The number of times routing tables are overflowed and the cause of overflow (DAO attacks).
- The number of RPL control messages sent and received (Resource depleting attacks).

In the following sections, we present the IDSs for RPL according to the IDSs' classification in Chapter 2-Section 2.4. Table 3.1 summarises the most relevant IDSs for RPL threats and categorises them according to detection method, used algorithms, addressed attacks, used dataset, evaluation metrics, and drawbacks.

### 3.7.3.1 Specification-Based IDSs

Le et al. [84] proposed a hybrid lightweight specification-based IDS idea for securing RPL against topology attacks. In this approach, nodes monitor routing information conveyed in control messages to detect attackers. As a continuation to this work, the authors implemented the proposed IDS using an Extended Finite State Machine (EFST) with statistic information about transitions and states for RPL (i.e., RPL normal profile) [118]. A cluster head requests its members to report its topology information periodically and process these pieces of information using EFST. Information used are: DIS sequence, number of DIS received, DIO sequence, number of DIO received, list of neighbours (i.e., Node ID, Rank, sequence of the DIO that provides this info, DIS sequence, number of DIS received, DAO sequence, number of DAO received, and a parent bit), and preferred parent ID. The IDS aims to detect Rank, sinkhole, local repair, neighbour, and DIS attacks. Zhang et al. [91] proposed a specification-based IDS with distributed Monitoring Nodes (MNs) to detect the routing choice attack. The detection data is network-based where in each MN is implemented a Finite-State-Machine (FSM) of RPL profile and used to detect intruders that deviate from RPL's normal behaviours. The attack is detected in the case when any malicious node multicast the DIO with lower ETX value, which consequently leads to a large fluctuation in the number of its child nodes than a set threshold. This node is marked as an attacker node. Surender et al. [119] proposed a constraint-based specification IDS for 6LoWPAN-based IoT networks to detect sinkhole attack (InDReS). This IDS depends on a FSM and behavioural rules to detect and isolate malicious nodes. In InDReS, sensor nodes are grouped into clusters with supervisor nodes. The latter track the number of dropped packets of their adjacent nodes and assign a score to each of them to detect malicious nodes. Once a malicious node is detected, it will be announced to all other nodes.

### 3.7.3.2 Signature-Based IDSs

Liu et al. [120] presented a signature-based IDS that uses Artificial Immune System mechanisms aiming to detect IoT attacks. The authors mapped the detectors to immune cells, antigens to the signature of datagram, malicious datagram to non-self-element, and normal datagram to self-element. In [121], a centralised signature-based IDS to detect DoS attacks in 6LoWPAN networks has been introduced. The proposed IDS has been integrated into the network framework ebbits developed within an EU FP7 project. In this IDS, non-6LoWPAN monitoring nodes located in the network send periodically the 6LoWPANs sniffed traffic through wired connection to the IDS. Thus, if a DoS attack occurs and degrades the wireless transmission quality, IDS data transmission would not be affected. The IDS collaborates with a DoS protection manager to confirm the attack using jamming information. Oh et al. [122] proposed a misuse-based IDS, which uses a lightweight pattern-matching algorithm, that has low computational complexity and requires small amount of memory to protect IoT networks. The authors suggested two matching techniques to match predefined attack signatures and IoT packet payloads, with the purpose to decrease the number of matches needed for detecting attacks. The proposed approach is faster than the Wu-Manber algorithm; one of the fastest pattern-matching algorithms.

### 3.7.3.3 Anomaly-based IDSs

A growing number of studies have been conducted to investigate anomaly-based IDS for IoT and especially RPL-based networks. Data mining, machine learning, statistical model, payload model, protocol model, rule model, and signal processing model are techniques that have been used in the literature for anomaly-based IDSs (see Figure 2.8). We classify the existing IDS for RPL as Machine learning (ML)-based, Deep learning (DL)-based and rule/statistical model-based.

**3.7.3.3.1 Rule/Statistical model-based IDSs** Raza et al. [58] introduced SVELTE, the first anomaly-based IDS for securing the RPL protocol. SVELTE modules were placed both in the BR and in the constrained nodes. At the first stage, the BR requests the network nodes to send information about themselves and their neighbours. These pieces of information are: RPL Instance ID, the DODAG ID, the DODAG Version Number, Rank, parent ID, neighbours list and their corresponding Ranks, and a timestamp. At the second stage, the BR analyses the collected data and makes decisions. SVELTE targets spoofed or altered information, sink-hole, and selective-forwarding attacks. It requires a low overhead to achieve a high detection rate. Pongle and Chavan [123] proposed an anomaly-based IDS to detect

packet relay and encapsulation types of wormhole attack, using Neighbor Discovery/Verification based techniques. Like SVELTE, the BR gathers information from constrained nodes to detect the attack. In this solution, four centralised modules are implemented on the BR and four distributed modules on the monitoring in-network nodes. The monitoring nodes gather information about their respective neighbours and changes on the network (RSSI), and send them to the BR. The latter analyses the received data to detect intruders and makes decisions. The simulation results showed that the energy overhead, the packet overhead, and the memory consumption were acceptable for constrained nodes. Cervantes et al. [124] proposed INTI, an IDS for sinkhole attacks over 6LoWPAN for IoT. INTI combines watchdogs, reputation and trust to detect sinkhole attackers. It organises the network on clusters where each node uses four modules to detect sinkhole attackers; cluster configuration, routing monitoring, attack detection and attack isolation. After monitoring the traffic, if a node detects an attacker it alerts other nodes to isolate the attacker. Thanigaivelan et al. [125] presented a cross-layer anomaly-based detection system for IoT. The proposed IDS is composed of three sub-systems located at the network and the link layers as follow: both the monitoring/grading subsystem (MGSS) and the reporting subsystem (RSS) operate at the network layer, whilst the isolation subsystem (ISS) operates at the link layer. If a node is detected to have abnormal behaviour, the ISS is used to avoid packets from that node at the link layer level. When anomalies and network changes are detected, they are communicated from the node to the edge-router through subsequent parents. The edge-router analyses reports and makes decisions. In particular, the approach monitors packet size and data rate. Gara et al. [126] introduced an anomaly-based IDS to detect the selective forwarding attack in IPv6-based Mobile IoT networks. This IDS works using two modules: a centralised module on the sink node (BR) and a distributed one on the routing nodes. Each monitoring node calculates periodically the number of packets received and the number of packets sent from each neighbour, and sends the collected data to the BR. These pieces of information are processed by the BR to detect malicious behaviour and decide whether a node is an attacker or not. If an attacker is detected, a global repair is triggered. In [127], a hybrid threshold-based IDS has been proposed to detect the DIS attack. The IDS uses the packet rate (i.e., DIS message sending rate) and the packet interval to detect the attack (see Section 3.6.2.1.5). In addition, the nodes' traffic is forwarded to the border router that will decide on the status of a node (i.e., malicious or not). An anomaly-rule-based lightweight IDS using threshold values has been proposed in [128] to deal with the neighbour and DIS attacks. Unlike the work cited above [118], the IDS is fully distributed in every node of the RPL network where each node monitors its neighbours to detect the attacks. A profile of normal behaviour for networks with different



sizes (i.e., 20, 30, and 40 nodes) has been defined. Every node stores the number of DIS messages received from its neighbours at specific time intervals. Afterwards, the maximum number of DIS messages received in all networks is calculated as the threshold to use. If the number of DIS messages received from a neighbour at specific time intervals is more than the threshold, that neighbour is considered as an attacker. Secure-RPL has been proposed in [97] to mitigate the effect of DIS flooding attacks. The Secure-RPL approach suggests discarding all DIS messages received before the expiry of the `RPL_CONF_DIS_INTERVAL` from a particular neighbour. Many data items such as sender IP address, previous DIS message receiving time, and the total number of DIS messages received since the last reset are collected and used to detect the intruder.

**3.7.3.3.2 ML-based IDSs** Sheikhan et al. [129] proposed a hybrid distributed IDS for real-time detection of the sinkhole, selective forwarding and Wormhole attacks using the NSL-KDD dataset. The model is based on the Map-Reduce approach that uses the Optimum-Path Forest (OPF), the Modification of Supervised Optimum-Path Forest (MOPF), and the Optimum Path Forest Clustering (OPFC), to classify nodes as normal or attacker. McDermott et al. [130] presented an experimental comparison of a Multi-Layer Perceptron Backpropagation Neural Network (BPN) and a Support Vector Machine (SVM) classifier to detect Denial of Service (DoS) attacks in WSNs using the NSL-KDD dataset. The authors concluded that both techniques offer a high true-positive rate and a low false positive rate, making both of them useful for intrusion detection. In [131], a compression header analyser based IDS (CHA-IDS) has been proposed to detect HelloFlood (i.e., DIS), sinkhole, and Wormhole attacks in an RPL network. The authors used Cooja-Contiki simulator to generate a dataset of 77 features. They used the Best First Search (BFS) and Greedy Stepwise (GS) to perform the features searching, then the Correlation-based Features Selection (CFS) algorithm to evaluate the most significant features. MLP, SVM, J48 (i.e., DT), NB, Logistic, and RF classifiers were compared and the results showed that J48 performs better than other classifiers for that specific configuration. Anthi et al. [132] proposed an IDS to detect network scanning probing and simple forms of DoS attacks in IoT networks. To generate the dataset, the authors used the software Wireshark to sniff network traffic. They tested several ML classifiers and used the NB classifier as it gave the best performance. Hasan et al. [133] presented a study comparing several ML methods to detect threats and attacks in IoT infrastructures. The authors concluded that RF classifier performs better than the other one. An open-source dataset of 13 features from kaggle4 was used. The dataset was gathered from a day of capture from the application layer using four simulated IoT sites. Although this work demonstrated the effectiveness

of RF classifier in the context of kaggle’s dataset, further studies need to take place to assess its performance for traffic from the network layer.

**3.7.3.3.3 Random Forests-Based IDSs for IoT** Primartha et al. [134] used three datasets, namely, NSL-KDD, UNSWNB15, and GPRS, to evaluate the performance of RF for IDS use. The authors assessed 10 RF classifiers with different number of trees (10, 40, 50, 80, ..., 800) and RF-800 gave better results. Comparing RF-800 to Random tree+NB tree, DMND, MLP, and NBTree, RF-800 outperforms the other classifiers with 99.57% for accuracy and 0.34% for false alarm rate. In [1], the authors proposed a cloud-based IDS using RF and Neural Network, and the UNSW-NB15 dataset. The IDS receives IoT traffic from the network device, performs features extraction, and classification on the extracted features. RF is used to detect if the data point is classified as an intrusion or not. Whereas, the Neural Network (i.e., one input layer, several hidden layers and one output layer) is used to categorise the detected intrusion. RF gives good results for precision, recall, and f1-score (99%, 98%, and 98%, respectively). Authors in [135] proposed TR-IDS, an IDS that uses word embedding and text-convolutional neural network (Text-CNN) techniques to extract features from the payloads in network traffic automatically, and RF for the classification. The authors used ISCX2012 dataset, from which they extracted 27 features to classify infiltration, BFSSH, HttpDoS, and DDOS attacks. They obtained the following performance: 99.13%, 99.26%, and 1.18% for accuracy, DR (Detection Rate), and false alarm rate, respectively. Tama et al. [136] proposed an IDS that uses particle swarm optimisation (PSO) for feature selection and RF classifier for attack detection. They used NSL-KDD dataset, where 37 features were selected to obtain an accuracy of 99.67%. The model outperformed rotation forest (RoF) and deep neural network (DNN) classifiers. The authors in [137] introduced AD-IoT, an RF-based IDS that monitors IoT traffic in a distributed fog layer to detect IoT Botnets at fog node, and alert the administrator. The authors used UNSW-NB15 dataset and ExtraTreesClassifier to reduce the number of features to 12. RF classifier achieved good performance with values of 99.34%, 98%, 98%, 98%, 0.2% for accuracy, precision, recall, F1-score, and false alarm rate, respectively.

**3.7.3.3.4 DL-based IDSs** One application of DL for intrusion detection in the IoT network is the work in [138]. The authors discussed the detection of Prob, DoS, U2R, and R2L attacks using fog-to-things architecture. They used NSL-KDD dataset with 128 features for detecting four classes of attacks. They also gave a comparison study of a deep neural network model with three hidden layers and a shallow neural network with as results 98.27% and 96.75% in term of accuracy, respectively. Authors in [139] have also applied a DL approach with five hidden

layers to detect RPL routing attacks. The authors generated datasets for decreased rank, HelloFlooding, and version number attacks relaying on several topologies. The obtained performance results in terms of F1-score for each dataset are 94.7%, 99%, and 95%, respectively. Qureshi et al. [140] proposed a deep random neural network-based heuristic intrusion detection system (RNN-IDS) for IoT. They trained RNN-IDS using the Gradient Descent Algorithm (GD). The authors used KDDTrain20 from NSL-KDD dataset to train the classifier with variant learning rates, and with both reduced features (29) and all features (41). The RNN-IDS accuracy reached up to 95.2% and gave better performance than SVM, NB, DT, MLP, and others.

### 3.8 Comparison and Discussion

As presented in the section above, several studies revolving around IoT and more specifically RPL security have attempted to design IDS systems tailored for it. Concerning the detection methodology, specification, signature and anomaly detections are deployed. Each method has its advantages and its drawbacks. The main and common drawback for the specification-based solutions is that human creates and develops protocol specifications, which means incorrect specifications, can result in false detections and might compromise the network. Besides, each solution has other specific disadvantages. For instance, the IDS proposed in [118] is energy efficiency but showed less accurate when it works for a long time. The one proposed in Zhang et al. [91] relies on unrealistic assumptions such as the stable state of LLN environment, secure network initialization, and static environment, which limits the practicality of the IDS. Although InDReS [119] performed well compared to INTI [124] in terms of packet drop ratio, PDR, control packet overhead, and average energy consumption, INTI considered only homogeneous nodes within a static network. In addition, the IDS may fail if the leader node itself is compromised. The above-solutions target specific attacks and are not capable of detecting new and unknown attacks in IoT. Moreover, if any situation invades the predefined system behaviour, the IDS decides that there is an intrusion.

Similar to the specification-based solutions, although signature-based IDSs have high detection accuracy and very low false alarm rate, they are not capable of detecting unknown attacks. In addition, it is complicated and time-consuming to build the signatures rules database since such IDSs are designed to detect malicious attacks and intrusions based on previous knowledge. Other common drawbacks for the signature-based solutions are the network packet overload and the challenge of frequently updating the signatures database, the high storage cost that grows with the number of attacks, and the requirement to compare the input with all the existing signatures. As already said, every solution has specific disadvantages. For

example, in [120], the authors did not discuss neither how the proposed immune-based IDS would be deployed in IoT networks, nor how the storage of signatures on low capacity IoT devices would be handled. The IDS proposed in [121] targets only DoS attack and is not compatible to a generic RPL architecture. It depends on developed rules to detect different other attacks and is not able to detect new and unknown attacks. The solution in [122] has several interesting features, such as the reduced memory size required for matching operations, the reduced workload for processing on smart objects, the increased speed of processing, and its scalable performance for a large number of patterns. Nevertheless, an attacker may try a unique pattern each time, making it difficult for the node to detect an attack.

As presented in the previous section, there exist several anomaly-based IDSs that have been proposed to detect intrusions against IoT and RPL-based networks. SVELTE [58] generates a high false positive, precisely when the number of attacks increases. Besides, SVELTE suffers from a synchronisation issue. For instance, the reported Rank information of a given node from the node itself and from its neighbours to the BR are not the same because the recording time was not synchronised. The IDS proposed in [123] targets only the wormhole attack and puts much communication and computational burden on resource-constrained nodes. Although INTI [124] is better than SVELTE [58] as it gives importance for node mobility and network self-repair, it targets mainly one attack, imposes extra network deployment cost, and the authors did not present the impact on the energy consumption. Moreover, the IDS placement change over the time can lead to consuming more resources. In the IDS proposed in [125] the parents themselves can be compromised and anomalies notifications can be avoided. Consequently, the solution may be not effective. In addition, no details were provided about the method of determining the normal behaviour of the network. Although the IDS in [126] deals with the mobility issue and offers good performance on attack detection, it generates high overhead, network congestion due to large number of ‘Hello’ packets and detects only two routing attacks.

The results from the IDS in [127] depend on the number of detectors within the network. Furthermore, the IDS introduces communication overhead where the higher the detectors, the higher the communication overhead. Besides, it is designed for static networks and the performance depends on threshold values. Likewise, the IDS in [128] deals with the DIS attack within a static network only and the performance depends on threshold values. The solution proposed in [97] has several drawbacks. A Sybil attacker can use different identities to avoid the mitigation mechanism. Moreover, even though all non-attacker nodes are configured with the same DIS interval, they need to be synchronised to detect the DIS attack (in which malicious node sends DIS after the expiry of DIS interval). Furthermore, the authors

tested their solution for small networks of 8 and 16 nodes with one attacker.

In our point of view, the IDSs mentioned above ([58][123][97][124][125][126][127][128]) are more likely seen as rule-based and/or statistic model-based IDSs since the BR gathers information from other nodes and decides the security status of the network relying on predefined rules, or nodes gather information from each other and detect the intruder relying on statistics and thresholds. Although these IDSs offer good performance on attack detection, they would generate high overhead and high false-positive rate if the number of nodes and/or attackers within the network increases. Besides, specific information need to be used to detect specific attacks. As a result, new rules need to be added to deal with unknown attacks.

One drawback for the solutions that use a threshold parameter to detect the routing attacks is how to set a threshold for different configurations and topologies, especially for a dynamic network? The second one is that several solutions assume that the attack is triggered after the DODAG stability is reached; however, an attacker can start the attack before the setup of the DODAG like a zero-day attack. While the nodes make statistics and count the number of specific messages to compare them with thresholds, the malicious nodes affect the performance of the network, which is another disadvantage for such solutions. Besides, the detection time related to the counting and the comparison with the threshold values will be higher with the growing size and the network's dynamic, which negatively influences the network's performance and goes against the real-time nature of the solution.

We suppose that ML and DL based solutions are more appropriate to handle some gaps of the above-cited solutions. It should be noted that even ML and DL IDSs have their own disadvantages in the specific case of RPL-based networks. For instance, the IDS presented in [129] has several drawbacks such as its high false positive and false negative rates compared to other IDSs in the literature. The authors consider a static network, which makes the results limited. Finally, despite the authors' goal to propose an IDS for real-time detection, they presented an execution time of 837 s (i.e., approximately 13 min), what goes against real-time characteristic. The IDS in [130] is evaluated using NSL-KDD dataset that is not specific to IoT networks. Even though the approach in [131] presents a good background for IoT ML-based IDS, the authors considered one topology and a small network of eight nodes. Furthermore, although the study in [132] is based on a testbed to generate datasets and detect intrusions, the authors used a small network composed of nine devices and the results present a low precision for DoS attack detection, which is not promising. The IDS in [133] targets threats from the application layer and further studies need to take place to assess its performance for traffic from the network layer. The works in [1], [134], and [135] are not specific to IoT networks, and the IDSs for IoT in [1] and [137] are evaluated using datasets

that are not specific to IoT and precisely to RPL-based networks.

The IDSs based on DL techniques such as [138] and [140] use NSLKDD, which is not specific to RPL-based networks. The work in [139] has fairly good results on an RPL-based datasets; nevertheless, the authors did not provide much detailed description of the simulation settings used to generate the datasets. In addition, they did not provide a multi-class evaluation.

As it can be noticed, most of the works deploy hybrid (hierarchical) architectures with a distributed network information gathering and/or analysis and a centralised detection. Furthermore, most of the existing IDSs have been tested on small network scenarios, but in the practical world, IoT is enabled by a large network of resource-constrained nodes. Therefore, the performance of the proposed IDSs may degrade in the case of large networks. Besides, almost all the proposed IDSs are designed for static networks. Besides the works in [139] and [131], one major drawback of the other works is using datasets from open-sources that are not designed explicitly for IoT and RPL-based networks. In addition, the studies did not use features that are relevant for RPL-based routing attacks detection.

A very important observation is the fact that all the proposed IDSs present solutions to detect intrusions against a network; nevertheless, no one present a solution to tolerate the intrusions.

Table 3.1: IDS solutions for RPL Networks 2011-2020 ... (Part 1)

Work	Detection Method	Algorithms/Description	Addressed Attack	Dataset	Metrics	Drawbacks
[120]	Signature	Artificial Immune System, r-Contiguous method to match signatures	-	-	-	The IDS deployment in the networks and signatures storage on low capacity IoT devices is not discussed.
[121]	Signature	Non-6LoWPAN monitoring nodes send periodically the 6LoWPAN sniffed traffic through wired connection to the Suricata IDS. The latter compares the traffic with the existing signatures.	DoS (IPv6 UDP flooding).	-	PenTest to evaluate the security. Number of true positive.	Although the solution does not depend on the network architecture, it targets only DoS attack and is not compatible to a generic RPL architecture. Depends on developed rules to detect different other attacks. Not able of detecting new and unknown attacks.
[58]	Anomaly	Collect and analyse routing information using rule-based algorithms.	Rank, sinkhole, and selective-forwarding.	-	True positive rate, Energy for entire network, Average power per node.	High false positive when the number of attacks increases. Synchronisation issue. Since IDS nodes use the network to transmit attack information, once DoS affects the network, it fails to detect DoS attack.
[122]	Signature	Lightweight pattern-matching algorithm using new auxiliary shifting and early decision techniques.	-	Snort and ClamAV pattern sets	-	An attacker may try a unique pattern each time, making it difficult for the node to detect an attack. Not able of detecting new and unknown attacks.
[91]	Specification	Collect and analyse routing information based on a Finite State Machine.	Routing choice intrusion.	-	Energy Efficiency, Delay, Extensibility, Fault Tolerant Ability.	Only as effective as the expertise level of the expert system. Considers attack defence only against a single intruder case. No performance analysis. Assumptions like secure network initialization, homogeneous nodes, monitoring nodes with more resources, and static environment limits the practicality of the IDS.
[123]	Anomaly	Collect and analyse routing information using rule-based algorithms.	Wormhole.	-	True Positive Rate.	Targets only the wormhole attack. Puts much communication and computational burden on resource constrained nodes.
[124]	Anomaly	Rule-based algorithms where each node monitors a number of transmissions performed by a superior node. If an attack is detected, an alert message is broadcast and a cooperative isolation of the malicious node is performed.	Sinkhole	-	Detection rate, False negative rate, False positive rate, Delivery rate of packets.	INTT's impact on energy consumption was not presented. IDS placement change over the time can lead to consuming more resources. Targets mainly mobile nodes and one attack. Imposes extra network deployment cost.
[84][118]	Specification	Collect and analyse routing information based on a Finite State Machine.	Rank, sinkhole, local repair, neighbour, DIS.	-	True Positive Rate, False Positive Rate, and Power consumption.	Less accurate when it works for a long time. Only as effective as the expertise level of the expert system. Performance depends on threshold values. RPL does not implement clusters.
[119]	Specification	Use of a Finite State Machine and behavioural rules to detect the intrusion.	Sinkhole	-	Packet Drop Ratio, Packet Delivery Ratio, Normalised Overhead, Throughput, Average Energy Consumption.	Only as effective as the expertise level of the expert system. Detects one attack.
[126]	Anomaly	Use of the Sequential Probability Ratio Test (SPRT) and the adaptive threshold of acceptable probability of dropped packets based on ETX to detect anomalies.	Sinkhole, selective-forwarding	-	Probability Detection, Total Overhead.	High overhead. Network congestion due to large number of 'Hello' packets. Detects only two attacks.
[129]	Anomaly	Optimum-Path Forest, Supervised Optimum-Path Forest, Optimum Path Forest Clustering, MapReduce	DR, SH, SF, Wormhole	NSL-KDD	Detection Rate, False Positive Rate, False Negative Rate	- Rises high false positive and false negative rates. No real-time detection as stated by the authors. Considers a static network.
[130]	Anomaly	MLP-Backpropagation-NN, Support Vector Machine (SVM)	DoS	False Positive Rate, True Positive Rate	NSL-KDD	NSL-KDD not specific to IoT/RPL networks.
[134]	Anomaly	RF, Random tree+NB tree, DMND, MLP, NBTree	DOS	NSL-KDD, UNSW-NB15, GPRS	Accuracy, False Alarm Rate, Significance Tests	Access the use of Random Forest Classifier for IDS use. Dataset not specific to IoT/RPL networks.
[131]	Anomaly	Best First Search (BFS), Greedy Stepwise (GS), Correlation-based Features Selection (CFS), DT, MLP, SVM, NB, Logistic, RF	HF, SH, Wormhole	Authors' dataset	Accuracy, False Positive Rate, True Positive Rate, Precision	One topology and a small network of eight nodes.

Table 3.2: IDS solutions for RPL Networks 2011-2020 ... (Part 2)

Work	Detection Method	Algorithms/Description	Addressed Attack	Dataset	Metrics	Drawbacks
[125]	Anomaly	Each node monitors its neighbour. If abnormal behaviour is detected, the monitoring node will block the packets from the abnormally behaving node at the data-link layer and reports to its parent node.	Packet Flooding, Selective forwarding, CloneID	-	False Positive Rate.	The parents themselves can be compromised and anomalies notifications can be avoided. No much details about the method of determining the normal behaviour.
[132]	Anomaly	Rule-based algorithm, NB	Quick/Quick Plus scan, Regular/Intense scan, SYN/UDP Flood	Authors' dataset	Precision, TPR, F1-score Precision, True Positive Rate, F1-score	Small network composed of nine devices. Low precision for DoS attack detection. Not delivering promising results for attack detection.
[1]	Anomaly	RF, DNN	-	UNSW-NB15	Precision, True Positive Rate, F1-score	Efficiency of detecting the category of the intrusion is insufficient. Dataset not specific to IoT/RPL networks.
[135]	Anomaly	Word Embedding, Text-Convolutional NN, RF	Infiltration, BFSSH, Http-DoS, DDOS	ISCX2012	Accuracy, Detection Rate, False Alarm Rate	Dataset not specific to IoT/RPL networks.
[136]	Anomaly	Particle Swarm Optimisation (PSO), RF, rotation forest (RoF), DNN	-	NSL-KDD	Accuracy, Precision, True Positive Rate, Significance tests	Dataset not specific to IoT/RPL networks.
[138]	Anomaly	DNN with three hidden layers	Prob, DoS, U2R, R2L	NSL-KDD	Accuracy, Precision, True Positive Rate, False Alarm Rate, F1-score	Using the NSLKDD Dataset that is not specific to IoT/RPL networks.
[139]	Anomaly	DNN with five hidden layers	Rank, HF, VN	Authors' RPL-based dataset (IRAD)	Accuracy, precision, recall, F1-score, False Alarm Rate	No multi-class classification. Low results compared to other IDSs.
[128]	Anomaly	Each node monitors its neighbours (statistics) and use a normal profile of the network and a threshold to detect the attack.	DIS, neighbour	-	True Positive Rate, False Positive Rate	Does not support mobility. - Performance depends on threshold values.
[133]	Anomaly	RF, LR, SVM, DT, ANN	DoS, Data Type Probing, Malicious Control, Malicious Operation, Scan, Spying, Wrong Setup	kaggle	Accuracy, Precision, True Positive Rate, False Positive Rate, F1-score	Dataset not specific to IoT/RPL networks. Further studies need to take place to assess its performance for traffic from the network layer.
[137]	Anomaly	ExtraTreesClassifier, RF	Botnets	UNSW-NB15	Accuracy, Precision, True Positive Rate, F1-score, False Alarm Rate	Dataset not specific to IoT/RPL networks.
[140]	Anomaly	Random-NN-Gradient Descent Algorithm, SVM, NB, DT, MLP, RF, RF Tree, Recurrent-NN, ANN	DoS, U2R, R2L, Probe	NSL-KDD	Accuracy, Precision, True Positive Rate, False Positive Rate	Dataset not specific to IoT/RPL networks.
[127]	Anomaly	Algorithms that use statistics on DIS messages and a threshold to detect the attack.	DIS	-	True positive rate, False positive rate, IDS warnings, Messages sent to IDS root	Depends on the number of detectors. The higher the detectors, the higher the communication overhead. Does not support mobility. Performance depends on threshold values.
[97]	Anomaly	Secure_RPL use statistics such as sender IP address, previous DIS message receiving time, and the total number of DIS messages received since the last reset to detect the attack.	DIS	-	Control packet overhead, Power consumption by radio transceiver	Performance depends selection of thresholds. A Sybil attacker can use different identities to avoid the mitigation mechanism. Nodes need to be synchronised to detect the DIS attack. Tested for small networks of 8 and 16 nodes with one attacker.



## 3.9 Summary

In this chapter, we provided an in-depth overview of the RPL standard operations. We outlined the RPL's topology construction while providing definitions of its control messages, routes construction, modes of operations, the standardised objective functions, and the most used RPL tools by the research community. We presented the RPL's security features and focused on the RPL's security gaps. We elaborated on how an adversary can exploit RPL vulnerabilities to trigger harmful attacks against the network and introduced a new classification of such attacks. We presented how the research community has responded to the RPL's security issues. Finally, we concluded the chapter with a discussion on the major drawbacks of the IDSs proposed to overcome the RPL's security gaps.

The first major identified gap in the literature review of RPL's vulnerabilities is the lack of RPL's performance evaluation under attack when the malicious nodes are mobile. In response, an analytical and a simulation-based evaluation of the RPL protocol under mobile Sybil attacks have been introduced in Chapter 4. In addition, a mitigation mechanism has been introduced.

## Chapter 4

# RPL's Performance under DIS and SybM Attacks and New Approach for the Intrusions Tolerance

As presented in Chapter 3, RPL is subject to several attacks that have been analysed for static networks. The research community did not consider the malicious node mobility. Nevertheless, IoT supports both static and dynamic-mobile applications. In response, this chapter introduces SybM, a DIS-Sybil attack against RPL with mobile Sybil nodes. We present an analytical and a simulation-based performance evaluations of RPL under SybM attack and a discussion on how the network performance can be affected. We propose a novel approach, namely RPL-MRC, to improve the RPL's resilience and thus its tolerance to both DIS and SybM intrusions. Finally, we assess the RPL-MRC performance under both static and dynamic RPL-based networks.

### 4.1 The Multicast DIS Attack (M-DIS): Reminder

RPL is based on the IPv6 Neighbour Discovery mechanism. It relies on Multicast operations to set up the network topology. A node within an RPL network sends a DIS message to solicit DIO messages from neighbouring nodes and join the DODAG. The DIS transmission interval varies from one RPL's implementation to another. In the RPL Cooja-Contiki simulator [82], it is handled using two constants; `RPL_CONF_DIS_START_DELAY` and `RPL_CONF_DIS_INTERVAL`. After a node starts (i.e., after booting), it delays the transmission of its first DIS message according to `RPL_CONF_DIS_START_DELAY`. A node aiming to join the network continuously transmits DIS messages within the `RPL_CONF_DIS_INTERVAL` fixed interval until it receives a DIO message from its neighbours. Upon receiving a DIO

message, it stops transmitting DIS messages and joins the network by sending a DAO message to its selected parent.

LLNs are not tamper-resistant, and nodes do not have a significant security defence. Hence, an adversary can compromise some nodes, reprogram and redeploy them into the network. As a consequence, even in the case of a secure RPL, the compromised nodes can use the pre-configured group key [22], and can normally participate in the network operations, and thus trigger attacks. In the M-DIS attack, the attacker exploits the RPL features mentioned above and frequently sends multiple Multicast DIS messages to its neighbours. Upon receiving a Multicast DIS message, the neighbouring nodes reset their DIO (Trickle) timers to the minimal value defined in the RPL implementation ( $2^{12}$  seconds) and send an excessive amount of Multicast DIO messages containing the up-to-date routing information [22], as illustrated in Figure 4.1. The repeated forced rest of the DIO timer leads to the wastage of the legitimate nodes' energy, and thus to shorten the network lifetime.

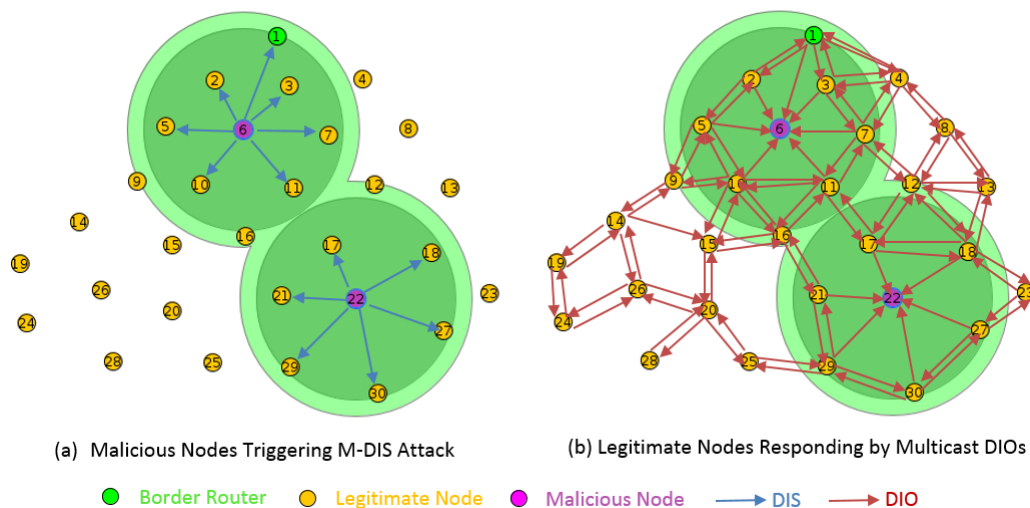


Figure 4.1: Multicast DIS attack illustration.

## 4.2 The Sybil Attacks

### 4.2.1 Sybil Attacks Overview

The most well-known definition of the Sybil attack is that a node claims multiple fake identities [141] called Sybil nodes [142]. The authors in [142] have analysed and classified Sybil attacks in sensor networks as follows.

- **Direct & Indirect Communication.** In the *direct* case, legitimate nodes communicate directly with Sybil nodes. In the *indirect* case, the communication is done through malicious nodes, which claim reaching the Sybil nodes.

- **Fabricated & stolen identities.** In the *fabricated identities* case, the malicious node creates a new identity for itself based on the identities of the legitimate nodes. For instance, if legitimate nodes have IPv6 addresses as ID, it randomly creates an IPv6 address. In the *stolen identities* case, the malicious node spoofs, and then uses identities of legitimate nodes which have been destroyed or disabled. For example, the attacker steals the IPv6 address of legitimate node.
- **Simultaneous & non-simultaneous.** In the *simultaneous* case, the malicious node involves all its Sybil identities in the network at the same time. In the *non-simultaneous* case, the malicious node alternately presents a subset of its identities over a period of time.

### 4.2.2 RPL under Sybil Attacks

The fact that RPL uses IPv6 addresses as nodes' identifiers, makes the routing protocol vulnerable to various Sybil attacks [24][141]. Once the adversary gains access to the network using a Sybil identity, it can exploit RPL other vulnerabilities to trigger different attacks. On the one hand, the malicious node can exploit the fact that RPL does not support mobility when routing to trigger a mobility-based attack. On the other hand, the adversary can exploit the functioning rules of RPL to trigger RPL related attacks (see Chapter 3). Both gaps can be combined and exploited by Sybil nodes to disturb RPL.

Sybil attacks are widely treated for different networks such as P2P, Ad-hoc and WSNs. Nevertheless, to the best of our knowledge, there are only few works that address Sybil attacks on RPL-based network without providing in-depth evaluation, which is worth to be investigated. For instance, in [143] and [144], the authors presented three types of Sybil attacks and their respective countermeasures. However, their analysis focus on SIoT (Social Internet of Things). The authors in [145] presented a method to detect a community of Sybil nodes. In [146], the authors presented a classification of existing Sybil attacks defence approaches, highlighting the effectiveness of LSD (Lightweight Sybil Attack Detection Framework). The aforementioned works stated that Sybil attacks are harmful for IoT networks, yet they did not present quantitative evaluation of the network especially for RPL performances. They focused only on Sybil detection approaches. In this chapter, we introduce three Sybil attacks against RPL and investigate the RPL's behaviour in the presence of one specific Sybil attack we named the Sybil Mobile attack (SybM for short).

For the proposed attacks, we assume each node can be mobile and automatically calculates its IPv6 address, which is used as identifier. It is assumed that in a

network of  $N$  nodes, an adversary can deploy  $M$  malicious mobile nodes (with  $M < N$ ). Each malicious node  $m$  has  $K$  Sybil identities. These identities are either fabricated IPv6 addresses or stolen identities. Each Sybil node  $i$  can join or leave the RPL topology any time.

**Sybm1.** In Sybm1 (Sybm for short) attack, the Sybil nodes communicate directly with legitimate nodes. They operate independently and do not cooperate during the attack. As depicted in Figure 4.2, in Sybm attack, each node is initially placed at a random location and sends periodically data packets to the BR. Malicious nodes pause for a period of time behaving the same way of honest nodes (sending data packets to the BR). Indeed, each adversary involves a set of its Sybil identities alternately and periodically, while moving through the network. Thus, after the pause time, malicious nodes choose a new location across neighbouring nodes towards the BR, and move physically there. When malicious nodes arrive, they repeat the process of pausing and then selecting a new destination to which they intend to move. Upon moving, malicious nodes multicast DIS messages within the network. In macro-mobility, normally, the IPv6 address of the node remains unchanged. Nevertheless, as in Sybm mobile nodes are malicious, they multicast DIS messages using new IPv6 addresses corresponding to new Sybil identities. The number of Sybil identities corresponds to the number of time an attacker moves. As a result, neighbourhood connectivity will change, and obviously more DIO messages will be exchanged.

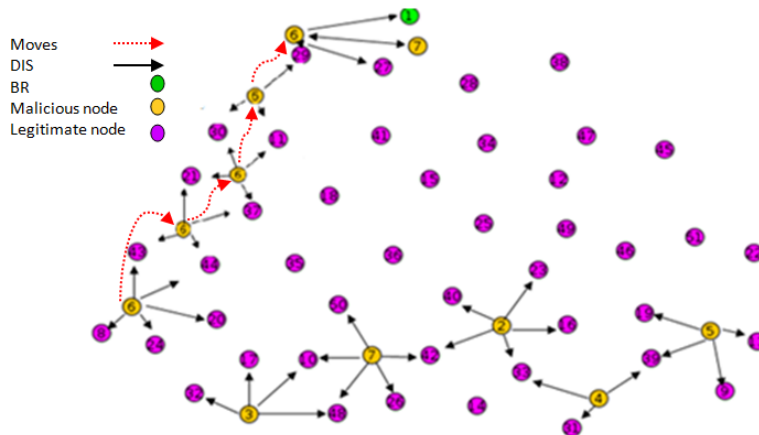


Figure 4.2: Sybm model, where 6 attackers move periodically across their neighbours towards the BR while multicast DIS messages.

**Sybm2.** Sybm2 is a Sybil attack based on stolen identities. RPL is based on IPv6, thus, it uses the IPv6's Neighbour Discovery mechanism, and Duplicate Address Detection Algorithm. However, the method for detecting duplicates is

not completely reliable, and it is possible that duplicate addresses will still exist [147]. Therefore, we can introduce a second Sybil attack, SybM2, exploiting this gap. In this attack, a node  $m$  moves near to a parent and uses its identity (i.e., steals the parent IPv6 address) to broadcast a poison, which is a DIO message with infinite Rank (i.e., Rank = INFINITE\_RANK). By sending a poison message, the parent (i.e., the malicious node) tells that it leaves the DODAG. As a result, the children of this parent could stay in its current DODAG through an alternative parent, which leads to the use of unoptimised route (increases packet delay), or might follow the leaving parent, which is more serious because it leads to break the sub-DODAG. When different attackers participate simultaneously in SybM2, damages on the network will be more important.

**SybM3.** SybM3 is triggered at the initial construction of the RPL topology. In this attack, the different attackers involve their different Sybil identities simultaneously (i.e.  $M * K$  Sybil identities at the same time). The Sybil nodes participate in the calculation of ETX aggregated metrics to choose the best and/or optimised route. This creates several wrong paths. As well, some correct routes can never be discovered. SybM3 leads to data loss or even an increase in packet delay.

### 4.3 Analytical-based RPL's Performance Evaluation under SybM Attack

In Section 4.2.2 we presented SybM, SybM2 and SybM3, and gave briefly the impacts of SybM2 and SybM3 on the network performance. On the one hand, SybM increases the DIO messages overhead, which is known to directly affect the energy consumption. On the other hand, the energy consumption is a critical concern for the IoT's devices. Therefore, as a first contribution, we provided an analytical evaluation of the RPL's performance under SybM attack [28]. Specifically, the impacts on messages overhead, packets delivery and energy consumption. Table 4.1 summarises the different notations used for the analysis.

In the analysis, we consider a network of 50 nodes ( $N = 50$ ). We consider different scenarios where the number of malicious nodes  $M$  increases from 2, 4, 6, 8 to 10, and for each malicious node  $m$ , the number of Sybil identities  $K$  increases from 1, 3 to 5 (i.e.,  $K$  Sybil nodes per attacker  $m$ ). Thus,  $M * K$  malicious mobile nodes will participate in the attack alternately for  $K$  cycles in groups of  $M$ . The lists of neighbours change from one Sybil node to another. We consider  $L_{mi}$ , the  $i^{\text{th}}$  Sybil

Table 4.1: Terminology

Notation	Description
Syb	Sybil attack over multiple cycles $\{1,2,3 \dots j \dots K\}$
j	$j^{\text{th}}$ cycle of the attack, which corresponds to the $j-1^{\text{th}}$ movement of an attacker
N	Number of nodes in the network
M	Number of mobile malicious nodes in the network
m	A mobile malicious node ( $m^{\text{th}}$ attacker)
K	Number of Sybil nodes per malicious node m (fabricated and/or stolen identities) which also corresponds to the number of cycles of Sybil attack
i	A Sybil node ( $i^{\text{th}}$ Sybil identity)
$L_{mi}$	Size of the $i^{\text{th}}$ Sybil node neighbors list -Number of neighbors of the $i^{\text{th}}$ Sybil node for the $m^{\text{th}}$ attacker- (corresponding to the $i-1^{\text{th}}$ move)
$L_m$	List of all neighbours for all Sybil nodes of one attacker m -Number of neighbours of the $m^{\text{th}}$ attacker for all its Sybil nodes- which corresponds to the number of neighbours for all $K-1$ moves
I, Imin, Imax	Trickle timer variables
DIO(j)	Number of exchanged DIO for cycle j of the attack
$N_{DIO}$	Number of exchanged DIO for the whole attack
$N_{DISsent}$	Number of transmitted DIS for the whole attack
$N_{DISreceived}$	Number of received DIS for the whole attack
$N_{DAOsent}$	Number of transmitted DAO for the whole attack
$N_{DAOreceived}$	Number of received DAO for the whole attack
$N_{DIOsent}$	Number of transmitted DIO for the whole attack
$N_{DIOreceived}$	Number of received DIO for the whole attack

node neighbours list of the  $m^{\text{th}}$  malicious node, is smaller when the attacker is closer to the leaf nodes or to the BR, and increases between them. Hence, we estimate for one attacker m an average sum,  $L_m$ , of all its Sybil-nodes neighbours will take values 4, 9 and 20 for 1, 3 and 5 Sybil identities, respectively.  $L_m$  is calculated following Equation 4.1.

$$L_m = \sum_{i=1}^K L_{mi} \quad (4.1)$$

### 4.3.1 Control messages overhead

When a Sybil node joins a DODAG, depending on its location, all or a part of the network topology will need to be updated. As a result, DIO messages will be exchanged more frequently between neighbours. Authors in [86] highlighted that,

when the number of attackers increases, control messages overhead increases too. In their experiments, they considered the DIS attack (see Section 4.1) where nodes are static. Under the constraint that the attacker nodes remain in the same location, their experiments show that, in a network of 50 nodes, the number of control messages doubles when the number of attackers is 4% of the total nodes, and increases 5 times more when the number of attackers is 20% of the total nodes.

We consider the DIS attack as a special case of the SybM attack. In fact, when nodes are static, the DIS attack corresponds to one cycle of SybM. In our analysis, we take DIS attack as the first cycle of SybM. Consequently, considering SybM, where  $M$  mobile malicious nodes move toward the BR and use their  $K$  Sybil identities to advertise repeatedly DIS messages, the number of control messages will increase substantially, exceeding the values reported in [86]. Indeed, for  $M$  Sybil nodes and cyclically (i.e.,  $K$  cycles), the topology has to be rebuilt almost from the BR towards the leaf nodes. We conduct our analysis on messages overhead on three steps. At the first step, we estimate the number of DIS and DAO messages exchanged during SybM as follows.

- Each malicious node  $m$  can send at least  $K$  DIS, and  $K$  DAO messages in broadcast transmission mode, using its different Sybil nodes. Based on the above assumption (i.e., receivers are listening at the time of sending without loss), we estimate that DIS and DAO messages are sent once per cycle  $j$  for each Sybil node  $i$ . In SybM there are  $M * K$  Sybil nodes; therefore, the numbers of DIS and DAO messages sent in the network are calculated following Equation 4.2.

$$N_{DISsent} = N_{DAOsent} = M * K \quad (4.2)$$

- Likewise, all the Sybil nodes neighbours receive both DIS and DAO messages in broadcast reception mode. Each node  $n \in L_{mi}$  neighbours list will receive one DIS at the beginning of the cycle  $j$ , and one DAO at the end. Therefore, the total number of DIS and DAO messages received for the whole SybM attack is equal to the number of neighbours directly affected by the  $M$  attackers, during the whole attack. Numbers of these messages are calculated following Equation 4.3.

$$N_{DISreceived} = N_{DAOreceived} = \sum_{m=1}^M \sum_{i=1}^K L_{mi} = \sum_{m=1}^M L_m = M * L_m \quad (4.3)$$

At the second step, we estimate the number of DIO generated by SybM. DIO messages are controlled using Trickle timer variables:  $I$ ,  $I_{min}$  and  $I_{max}$  with  $I \in [I_{min}, I_{max}]$ . When the topology is not in a steady state (i.e., a node joins the network),



nodes will reset the timer to  $I = I_{\min}$  [22], and send more DIO messages. From one cycle  $j$  to another, malicious nodes move towards the BR. As a consequence,  $I$  decreases and the number of DIO messages ( $DIO(j)$ ) increases. The total number  $N_{DIO}$  of DIO messages exchanged during SybM is the sum of the exchanged  $DIO(j)$  messages for each cycle  $j$  of the attack (see Equation 4.4).

$$N_{DIO} = \sum_{j=1}^k DIO(j) \quad (4.4)$$

$N_{DIO}$  includes DIO messages in broadcast transmission and reception. When a Sybil node sends a DIS message, all its neighbours are affected. An affected node  $n$  (which is not a leaf node), upon receiving DIS or DIO messages, sends in broadcast transmission one DIO message to its neighbours, and receives in broadcast reception DIO messages from all its neighbours (i.e., transmitting nodes). The process is repeated for all neighbours. For example, as seen in Figure 4.3, in the network there are 4 leaf nodes and 6 transmitting nodes. Thus, there are only 6 DIO messages sent and 18 DIO messages received (i.e., number of arrows), which represents  $\frac{2}{3}$  of the exchanged DIO messages. We have selected several scenarios to estimate the ratio between  $N_{DIO_{sent}}$  (i.e., the number of DIO sent) and  $N_{DIO_{received}}$  (i.e., the number of DIO received) during RPL topology reconstruction. Based on our benchmark results, we could assume that  $N_{DIO_{sent}}$  represents approximately  $\frac{1}{3}$  of  $N_{DIO}$  and  $N_{DIO_{received}}$  represents approximately  $\frac{2}{3}$  of  $N_{DIO}$ . Therefore, In our analysis, we rely on Equation 4.5 and Equation 4.6 to calculate  $N_{DIO_{sent}}$  and  $N_{DIO_{received}}$ , respectively.

$$N_{DIO_{sent}} \approx \frac{1}{3} * N_{DIO} \quad (4.5)$$

$$N_{DIO_{received}} \approx \frac{2}{3} * N_{DIO} \quad (4.6)$$

Finally, the total control traffic overhead is calculated following Equation 4.7.

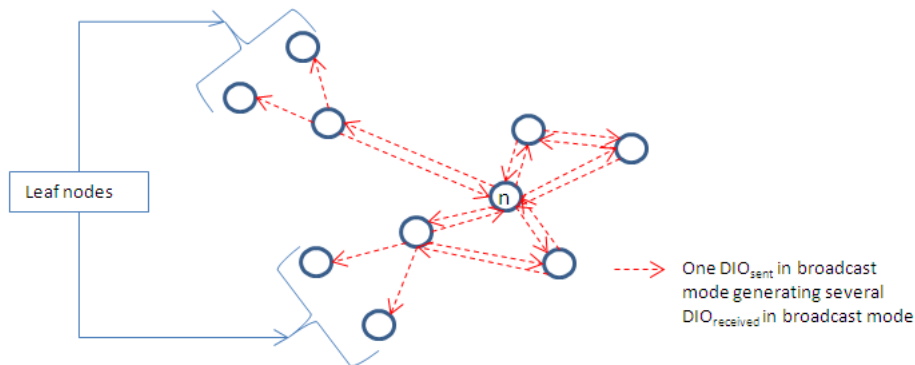


Figure 4.3:  $N_{DIO_{sent}}$  and  $N_{DIO_{received}}$  within the network.

$$Control\_traffic\_overhead = N_{DIOsent} + N_{DISsent} + N_{DAOsent} + N_{DIOreceived} + N_{DISreceived} + N_{DAOreceived} \quad (4.7)$$

Figure 4.4 represents DIS and DAO messages overheads for our different scenarios. The number of DIS messages is calculated as the sum of  $N_{DISsent}$  and  $N_{DISreceived}$ . Likewise, the number of DAO messages is calculated as the sum of  $N_{DAOsent}$  and  $N_{DAOreceived}$ . As well, Figure 4.5 and Figure 4.6 represent DIO messages and control traffic overhead, respectively.

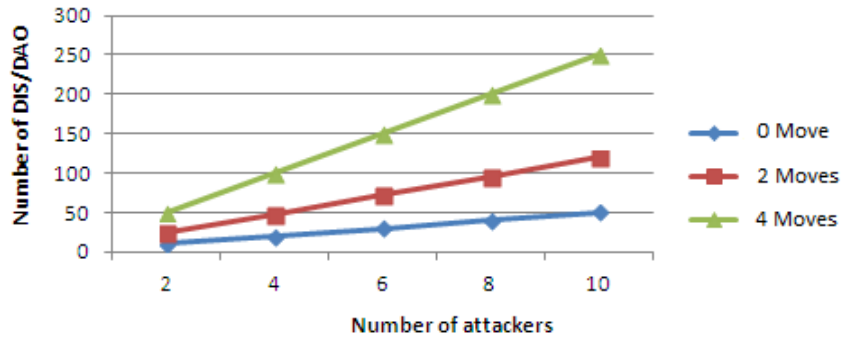


Figure 4.4: DIS/DAO messages overhead.

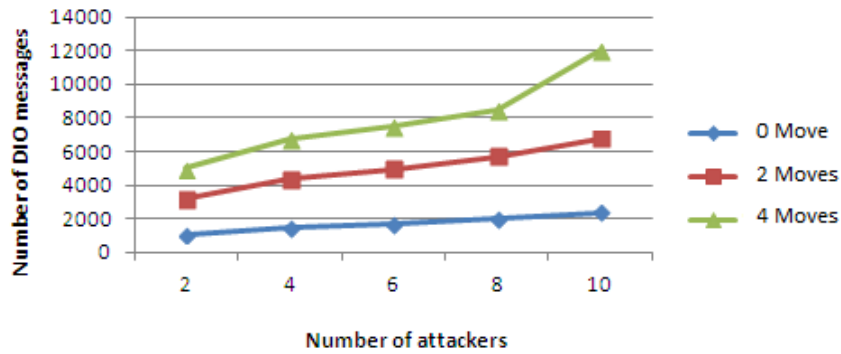


Figure 4.5: DIO messages overhead.

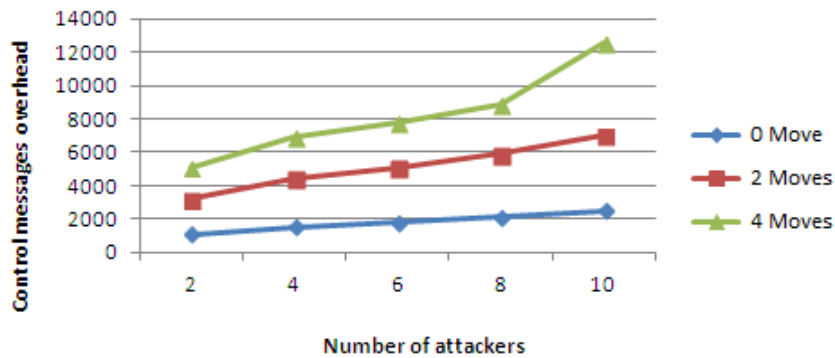


Figure 4.6: Control messages overhead.

### 4.3.2 Packets delivery

For packet delivery, we can highlight two cases. In the first case, there is no limitation on the number of joining nodes; the network is never in a steady state. In this case, control messages will take an important part in the network data flow. Therefore, data packets transmitted will be hindered due to their longer waiting in queues, which leads to their removal, and hence to a denial of service. In the second case, there is configured limitation on the number of joining nodes (e.g., only 20 joining requests per 1h). In this case, the network will be periodically stable allowing more data packets transmissions. Nevertheless, data packets could be lost in two possible ways. The first one, if they travel through Sybil nodes, which are moving. The second one, if the Sybil nodes deliberately remove them. As a result, the number of packet loss will increase.

### 4.3.3 Energy consumption

The energy consumption is directly related to control messages overhead. Hence, we have to calculate the energy cost of DIO, DIS and DAO overheads to estimate the energy consumption caused by SybM. For this purpose, we rely on messages overheads calculated following Equations 4.2, 4.3, 4.4, 4.5 and 4.6, and on the energy model of TelosB constrained nodes defined in [9], and summarised in Table 4.2. In this model, a node consumes 178  $\mu\text{J}$  for one broadcast message reception, and

Table 4.2: Energy consumption on constrained node [9].

Activity	Energy ( $\mu\text{J}$ )
Broadcast reception	178
Broadcast transmission	1790

1790  $\mu\text{J}$  for one broadcast message transmission. In Table 4.3 and Table 4.4, we summarise the energy cost for two scenarios (i.e., the first one when  $K=5$  and  $M=10$ , and the second one when  $K=1$  and  $M=10$ ) for each control message type (DIO, DIS and DAO), and the total energy consumption due to SybM.

Table 4.3: Energy cost for SybM ( $K=5$  and  $M=10$ ).

Control Type Activity	DIO	DIS	DAO	Total ( $\mu\text{J}$ )
Broadcast reception	8000*178 1424000	200*178 35600	200*178 35600	1495200
Broadcast transmission	4000*1790 7160000	50*1790 89500	50*1790 89500	7339000
Total( $\mu\text{J}$ )	8584000	125100	125100	8834200

Table 4.4: Energy cost for SybM (K=1 and M=10) corresponding to the energy cost of DIS attack.

Control Type \ Activity	DIO	DIS	DAO	Total ( $\mu\text{J}$ )
Broadcast reception	1600*178 284800	40*178 7120	40*178 7120	299040
Broadcast transmission	800*1790 1432000	10*1790 17900	10*1790 17900	1467800
Total( $\mu\text{J}$ )	1716800	25020	25020	1766840

Table 4.5 and Figures 4.7 and 4.8 illustrate the energy cost generated by all presented scenarios of SybM.

Table 4.5: Energy cost (J) for SybM attack depending on numbers of Sybil nodes and attackers.

K \ M	2	4	6	8	10
1 (DIS attack)	0,782565	1,099454	1,274704	1,48429	1,76684
3	2,28548	3,17606	3,63744	4,206264	4,9894
5	3,6274	4,92858	5,51512	6,279778	8,8342

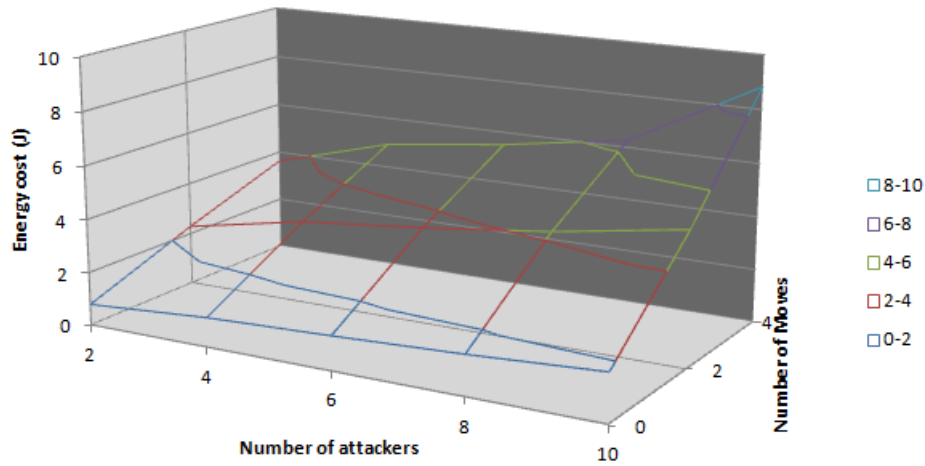


Figure 4.7: SybM's energy cost.

#### 4.3.4 Discussions

Figures 4.5, 4.6 and 4.8 show that DIO, control overhead and energy cost evolve similarly. We notice that the energy cost increases in the same way as control messages overhead. Also, the energy cost increases in the same way as DIO messages overhead.

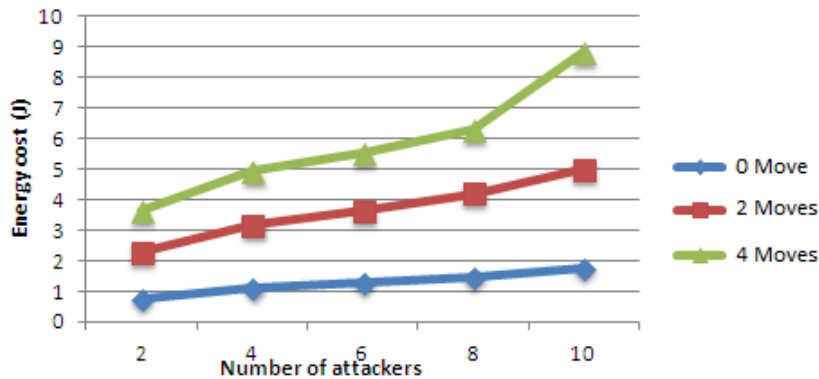


Figure 4.8: SybM's energy cost for  $K=1$ ,  $K=3$  and  $K=5$ .

According to Tables 4.3 and 4.4, the highest rate of energy consumption is caused by DIO messages, which represents about 97% of the total energy consumption. Furthermore, it can be seen clearly in Figures 4.4, 4.5, 4.6 and 4.8, where blue curves (i.e.,  $K=1$ ) represents the evolutions of messages overhead and energy cost for DIS attack, that the estimated messages overhead and the energy cost caused by SybM exceed by almost  $K$  times (i.e., number of Sybil identities per attacker) the ones caused by the DIS attack. Moreover, as seen in Figures 4.5, 4.6, 4.7 and 4.8, messages overhead and energy consumption increase steadily when the number of Sybil nodes is less or equal to 3 (i.e.,  $K \leq 3$ ), or when the number of Sybil nodes exceeds 3 and the number of attackers is less or equal to 8 (i.e.,  $K > 3$  and  $M \leq 8$ ). However, when the number of attackers exceeds 8 (i.e.,  $K > 3$  and  $M > 8$ ), messages overhead and energy consumption increase considerably compared to the other cases. Likewise, according to Table 4.5 and to Figures 4.4, 4.5, 4.6, 4.7 and 4.8, messages overhead and energy cost almost double when the number of Sybil nodes increases (the gap in the figures between  $K=3$  and  $K=5$ ). For 10 attackers, the DIS attack (i.e.,  $K=1$ ) causes an energy consumption of 1,76684J. However, SybM with 5 Sybil identities (i.e.,  $K=5$ ) causes an energy consumption of 8,8342J. Thus, there is an energy cost difference of approximately 7J between the two attacks, which significantly reduces the lifetime of the network. These observations are due to the fact that the number of Sybil nodes becomes too close or equal to the number of the nodes in the network, which means that almost the whole topology is affected.

## 4.4 Simulation-based RPL's Performance Evaluation under SybM Attack

Our first contribution and other works in the state-of-the-art stated that Sybil attacks are harmful for IoT networks, but they did not present quantitative evalu-

ation of the RPL network performance. In our second contribution, we present a simulation-based study and analysis of RPL's performance under the SybM attack.

#### 4.4.1 Simulation Settings

We simulated a network of 50 TelosB nodes (Sky motes) with one BR placed in the centre and 49 senders placed randomly around the BR. Each Sky mote is powered by an 8MHz, 16-bit Texas Instruments MSP430 microcontroller with 10kB of RAM and 48kB of flash memory. Table 4.6 shows the simulation parameters.

Table 4.6: Simulation parameters for SybM attack's effects on RPL

Parameter	Value
<b>Simulator</b>	Cooja-Contiki 2.7
<b>Simulation time (s)</b>	330
<b>Number of nodes</b>	50
<b>Network area</b>	$300 \times 300 m^2$
<b>Transmission range</b>	50m
<b>Radio medium</b>	UDGM : Distance Loss
<b>Traffic rate</b>	1 packet sent every 10 seconds
<b>Number of mobile/attacker nodes</b>	0, 2, 4, 6, 8, 10
<b>Number of Sybil nodes per attacker</b>	1, 3, 5

We simulated four scenarios as summarised in Table 4.7. The first and second scenarios are used as benchmarks. As the DIS attack represents a special case of SybM where attackers are static nodes, the third scenario represents the implementation of the DIS attack and is also used as a benchmark [86]. The fourth scenario represents the SybM attack.

We conducted the simulations on Cooja-Contiki-2.7. To handle nodes mobility, we used the Cooja-Mobility-Plugin. Furthermore, to handle Sybil identities we rely on the work in [148]. For more accurate evaluation, each simulation was executed 5 times with random seeds and simulations outputs were averaged. To study the impacts of SybM attack on RPL performances, we focused on control messages overhead, packets delivery and energy consumption parameters. For the control messages overhead and the energy consumption analysis, we used the radio messages and collect-view tools from Cooja. For the packets loss analysis, we used the simulation script editor from Cooja. Figure 4.9a represents the experimental network topology for SybM attack in the case of 10 malicious nodes before triggering the attack. Whereas Figure 4.9b represents the topology evolution of the same network after triggering the attack. The mobility issue is seen clearly even without attacker. Once the node 28 moves, the node 45 becomes isolated from the network.

Table 4.7: Scenarios

Scenario	Description
<b>First</b>	A network with no attacker and no mobility
<b>Second</b>	A network with no attacker and some mobile nodes. We varied the number of mobile nodes from 2, 4, 6, 8, to 10. Each mobile node moves towards the BR 1, 3 then 5 times (noted 1Move, 3Move and 5Move, respectively). The special case of 0 mobile node and 0 move corresponds to the first scenario
<b>Third</b>	A network with attackers (Sybil nodes) and no mobile nodes. In this scenario the attackers multicast periodically DIS messages from the same locations. We varied the number of attacker nodes from 2, 4, 6, 8, to 10. For each attacker the number of Sybil identities increases from 1, 3, to 5 (noted 1DIS, 3DIS and 5DIS, respectively)
<b>Fourth</b>	SybM attack scenario. We varied the number of Sybil mobile attacker from 2, 4, 6, 8, to 10 attackers. Likewise, the number of Sybil identities per attacker increases from 1, 3, to 5 (noted 1SybM, 3SybM and 5SybM, respectively)

## 4.4.2 Simulations Results

### 4.4.2.1 Control Overhead

Figure 4.10 depicts the control overhead under SybM attacks and the first scenario (i.e., no attack). When we compare SybM with one Sybil node per attacker (i.e., 1SybM) with the No-attack scenario, we notice that the extra control overhead after triggering 1SybM is about 6% for 2 moving attackers, and increases until reaching 32 % for 10 moving attackers. Likewise, for SybM with 3 Sybil nodes per attacker (i.e., 3SybM), the extra control overhead is about 24% for 2 moving attackers, and increases until reaching 66% for 10 moving attackers. In the case of SybM with 5 Sybil nodes per attacker (i.e., 5SybM), the extra control overhead is about 45% for 2 moving attackers, and increases until reaching 133% for 10 moving attackers. In the case of 1SybM and 3SybM, when the number of mobile attackers increases, the control overhead increases steadily, while it increases considerably in the case of 5SybM until being doubled. Furthermore, by increasing the number of Sybil mobile nodes within the network, the control overhead increases significantly. The extra control overhead from 3SybM is 2 times the one from 1SybM (in the case of 4 and 6 moving attackers the overhead almost doubles). In addition, the extra control overhead from 5SybM is almost 2,5 times the one from 3SybM (in the case of 8 and 10 attackers the overhead exceeds the double).

In the second scenario (see Figure 4.11), we notice that even when the number of moving nodes increases, the overhead generated by 1 or 3 moves per moving node remains almost the same. However, when the number of moves exceeds 3 (i.e.,

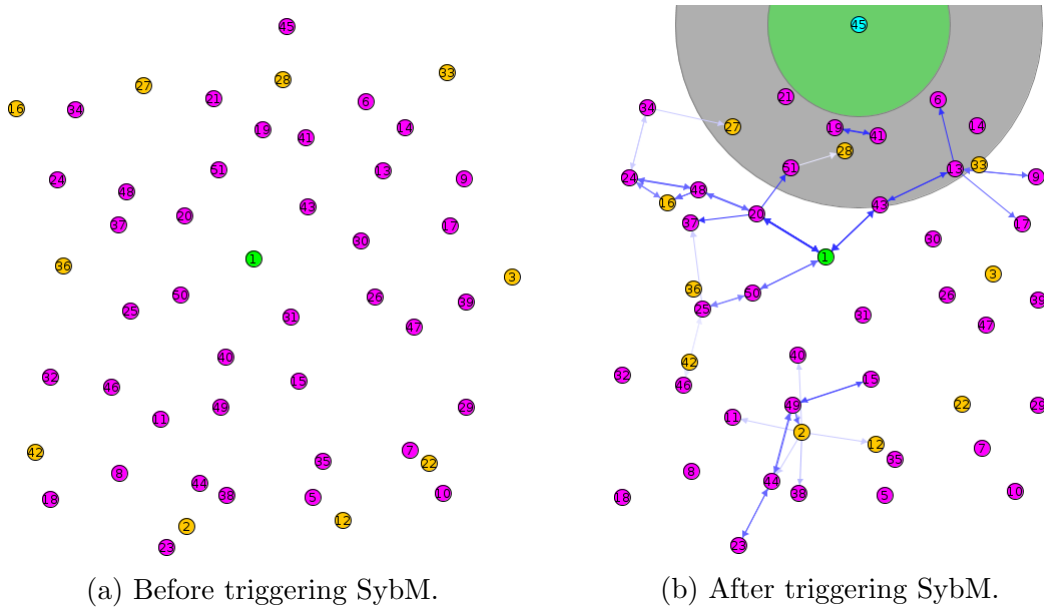


Figure 4.9: The experimental network topology under SybM attack in the case of 10 malicious nodes.

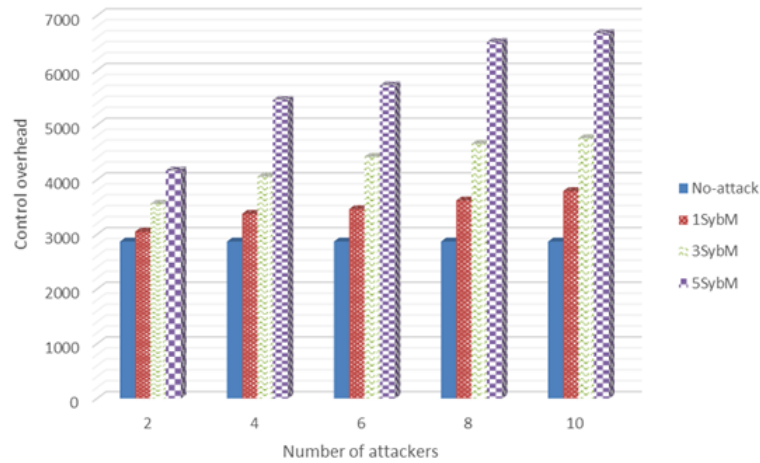


Figure 4.10: SybM attacks control overhead

is equal to 5), the overhead is more significant. It is almost the same overhead for 1SybM. Moreover, in the second and fourth scenarios, when the number of moves/Sybil-nodes exceeds 3 (i.e., 5Move and 5SybM cases) the control overhead increases because mobile and Sybil nodes are more close to the BR, and thus can be detected by it; which means the whole DODAG needs to be reconstructed from scratch. Furthermore, even if the mobile nodes in the two scenarios move in the same way (same positions), we notice that the overhead generated by SybM is almost twice the second scenario. This is due to the nature of SybM and the RPL Trickle timer mechanism. In the second scenario there is no mechanism to detect mobile nodes and thus, the Trickle timer interval is not updated accordingly. Nevertheless, in SybM scenario, in addition to mobility, submission of DIS messages from different



locations resets the Trickle timer and fasten the exchange of more control messages.

In Figure 4.11, we see that in both the third and fourth scenarios, when the number of Sybil nodes increases, the extra overhead increases too. Likewise, when the number of malicious nodes increases, the extra overhead increases too. Indeed, when the number of Sybil nodes increases, in the case of 2 attackers, the extra overhead of Sybm almost doubles compared with the DIS attack. Furthermore, in the case of 10 attackers, the extra overhead of Sybm almost triple compared with the DIS attack. This is due to the fact that DIS attack (i.e., 0 move) represents a static environment while varying the number of attackers. Whereas, Sybm attack represents a dynamic environment where the mobility of malicious nodes causes the number of nodes affected by the attack to increase, and hence, the control overhead as well. For Sybm attack, we notice that the overhead is almost the same in the case of 8 and 10 attackers. Besides, it is almost the same in the case of 4 and 6 attackers. This can be explained by the fact that attackers are moving almost in the same area, and thus, affect the same neighbouring nodes. In addition, the attackers can be close to the BR, which involves reconstructing the whole topology. As seen in Figure 4.9b, from 6 attackers the node 45 is completely isolated and do not participate any more in the network. This also partly explains why the overhead do not increase as expected when increasing the number of attackers.

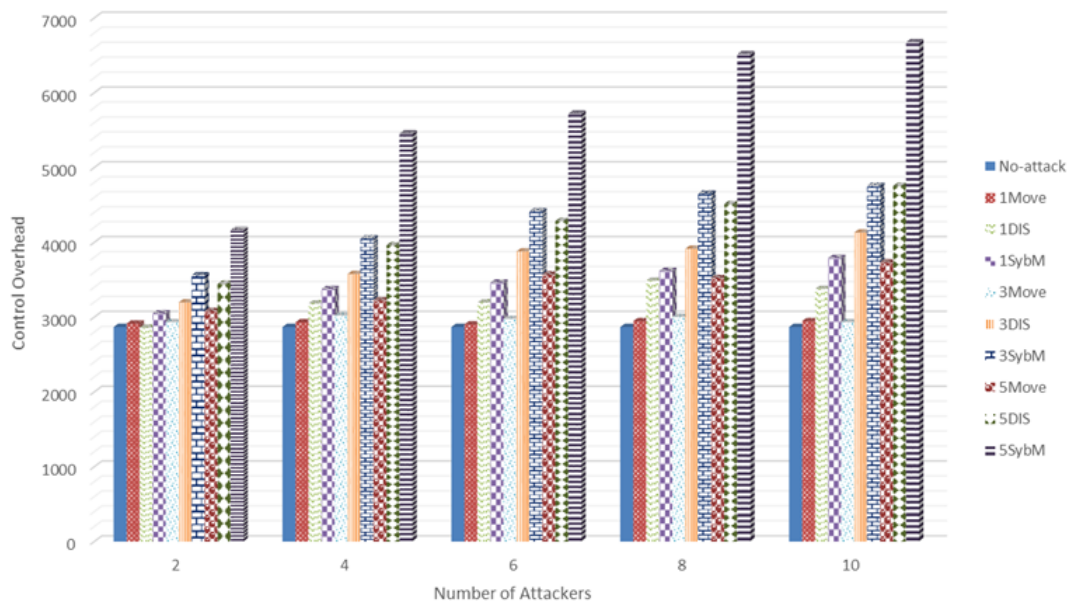


Figure 4.11: Control overhead vs number of attacker

#### 4.4.2.2 Packets Delivery and Energy Consumption

In Figure 4.12 and Figure 4.13, we see that in the presence of SybM attackers, the energy cost increases whilst PDR reduces remarkably, as the number of attackers and Sybil nodes increase. This could be due to the growth of affected nodes within the network. Consequently, the number of exchanged control messages is increased, which rises the probability of collisions and packets retransmission, and in turn increases the power consumption and lowers the PDR. In addition, we notice that damaging effects from SybM in terms of energy cost and PDR outpace the one from DIS attack by up to 33%. The effect of DIS attack on PDR is smaller even when compared with the second scenario. This is because in DIS attack, nodes are not mobile, and thus only few packets will be lost due to probable collisions. However, In the case of SybM and even in the second scenario, packets sent to mobile nodes will systematically be lost if nodes are moving, which reduces PDR. On the other hand, the energy cost occasioned by DIS attack is more important than the one in the second scenario. This is due to the fact, in DIS attack there is more control overhead and thus more energy consumption. PDR is more important in SybM scenarios and in the second scenario because of the RPL's mobility handling gap.

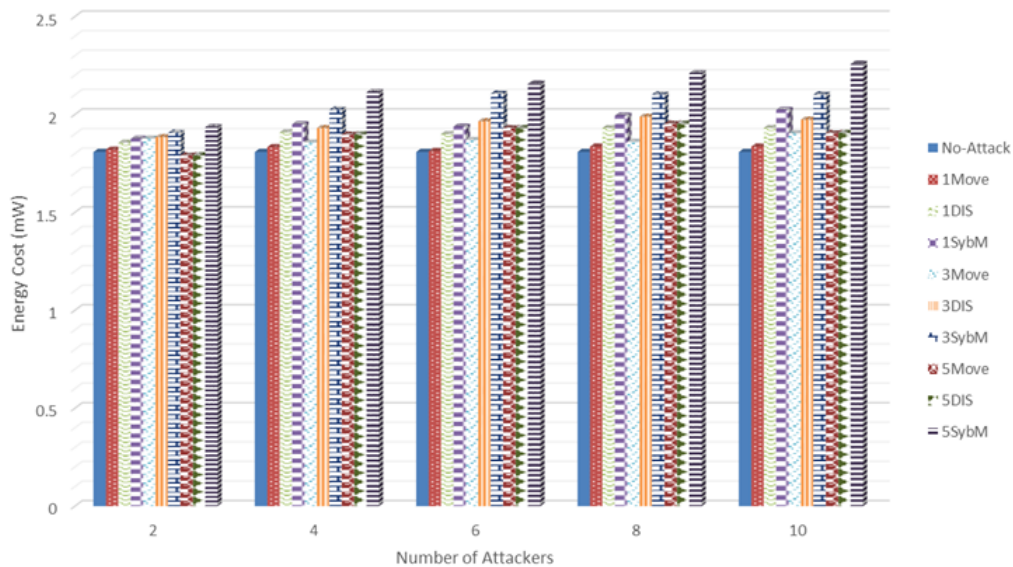


Figure 4.12: Energy cost vs number of attacker

## 4.5 The Proposed Approach: RPL-MRC

As it can be seen in Table 3.1, a few solutions have been proposed in the literature to deal with the DIS attack. One drawback for the solutions that use a threshold parameter to detect the DIS attack is how to set a threshold for different configurations and topologies, especially for a dynamic network. The second one is that

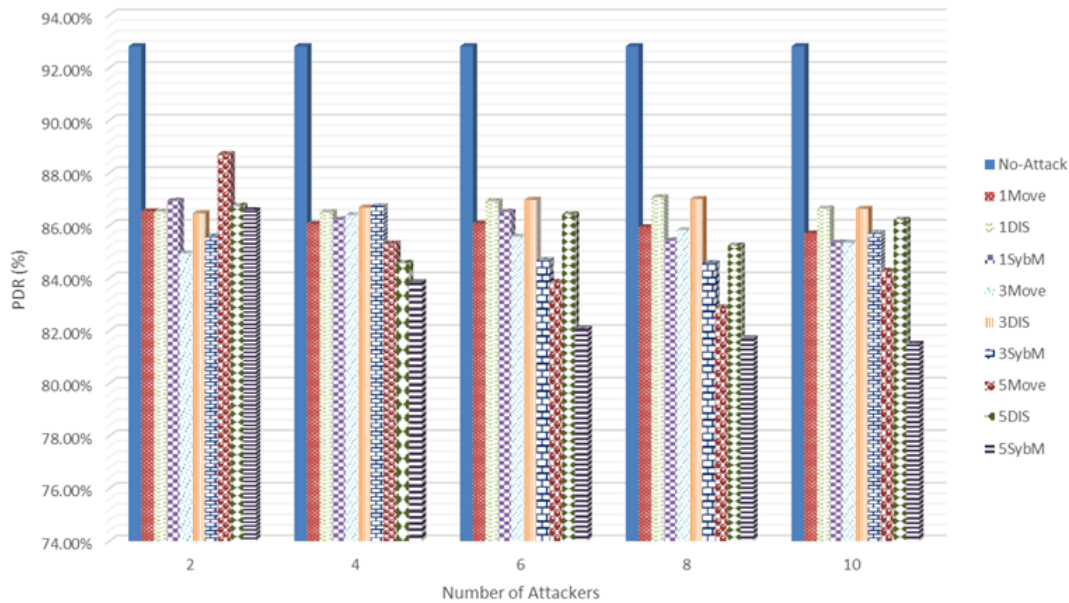


Figure 4.13: Packet delivery ratio vs number of attacker

several solutions assume that the attack is triggered after the DODAG stability is reached; however, an attacker can start the attack before the setup of the DODAG. While the nodes count the number of DIS messages to compare it with a threshold, the malicious nodes affect the performance of the network, which is another disadvantage for such solutions. Besides, the detection time (related to the counting and the threshold) will be higher with the growing size and the network's dynamics.

To address the M-DIS and SybM attacks and remedy the aforementioned shortcomings, we propose a new mitigation and intrusion tolerance approach that is composed of two complementary mechanisms: Response Delay and Timer Readjustment. We integrate the Response Delay mechanism into the `dis_input` function of the RPL implementation, whereas the Timer Readjustment is integrated into the `new_dio_interval` function responsible for the timer's reset.

### 4.5.1 Response Delay

With RPL-MRC, RPL itself is adapted to reduce the response to Multicast messages, thus reducing the impact of the attacks on RPL-based LLNs. RPL-MRC is inspired by the Multicast Listener Queries (MLQ) principle described in the RFC 3810 [149]. Multicast routers send MLQ Messages in Querier State to query the multicast listening state of neighbouring interfaces. In the Queries format, a two-bytes field named the Maximum Response Code (MRC) specifies the maximum time allowed before sending a responding Report. It represents a floating-point value. The actual permitted time to respond is called the Maximum Response Delay (MRD). MRD is expressed in units of milliseconds and is derived from the MRC.

As the MLQ messages presented in RFC 3810, RPL-MRC uses an MRC field to reduce responses to Multicast DIS messages. To this end, we redefined the DIO Base Object as follows. We use the one-byte Reserved field as an MRC field set by the border router, as depicted in Figure 4.14. The MRC value must be greater than the  $I_{min}$  value of the Trickle timer and smaller than the  $I_{max}$  value (i.e.,  $I_{min}$  plus the redundancy value  $k$ ), as defined in Chapter 3, Section 3.3. On receiving a Multicast DIS message, instead of responding immediately with a Multicast DIO message, the legitimate node delays its response by a random amount of time in the range  $[MRD/2, MRD]$ , where the MRD value is calculated as in Equation 4.8. The value 3 is chosen according to several simulations that demonstrated its effectiveness in tolerating the intrusions. In addition, we restricted the number of Multicast responses as follows. While delaying the response, every node tracks the number of DIO messages responding to the DIS Multicast. Suppose their number exceeds a pre-specified threshold less than the one allowed by the Trickle timer (i.e., the redundancy variable defined in Chapter 3, Section 3.3). In that case, the node cancels its pre-programmed response.

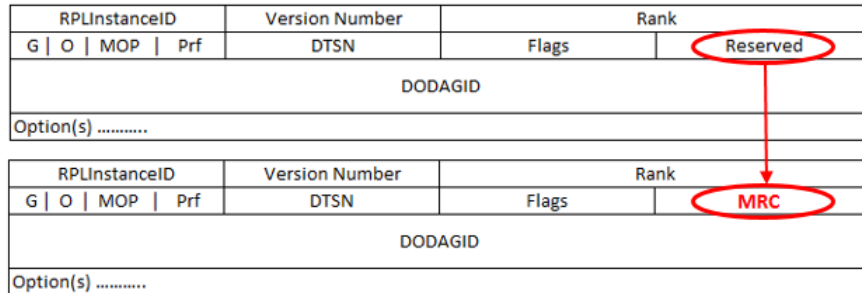


Figure 4.14: New DIO Message

$$MRD = \begin{cases} 2^{MRC} & \text{if } I_{min} < MRC < I_{min} + k, \text{ and } k > 3 \\ 2^{I_{min}+3} & \text{else.} \end{cases} \quad (4.8)$$

### 4.5.2 Timer Readjustment

As explained in Chapter 3 (Section 3.3), the Trickle algorithm involves three configuration parameters and three variables to govern transmission of the control traffic used to construct and maintain the DODAG. The idea behind the timer is to adjust and regulate the frequency of DIO messages transmission based on network conditions. Firstly, the timer changes adaptively the transmission rate where it increases the transmission rate when a change in routing information is discovered (i.e., receiving a DIS message, a new version number, changes in the link layer quality, etc.) in order to propagate up-to-date information rapidly. As the network approaches its steady phase, the timer exponentially reduces the transmission rate as there is no

update to propagate. Secondly, the timer uses a suppression mechanism in which a node suppresses the transmission of its control packet if it detects that enough of its neighbours have transmitted the same piece of information, thus limiting redundant transmissions.

When a node receives a Multicast DIS message, and the transmission period  $I$  is greater than  $I_{\min}$ , it terminates (i.e., suppresses) the scheduled transmission of DIO messages (i.e., at the current sub-interval  $I$ ). It reinitialises the DIO Trickle timer from a sub-interval of a minimum length (i.e., sets  $I$  to  $I_{\min}$ ), as shown in Figure 4.15. If  $I$  is equal to  $I_{\min}$  when the node hears the Multicast DIS transmission, it does nothing (i.e., it waits for the scheduled DIO at time  $t$ ) [75].

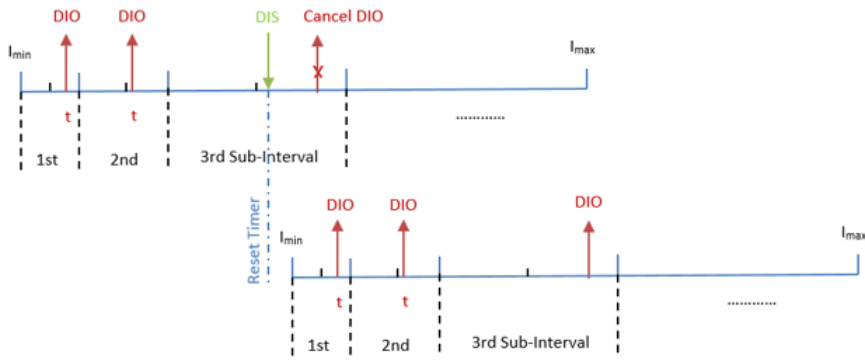


Figure 4.15: The Trickle Timer on Receiving Multicast DIS Message

With the RPL-MRC approach, the node reinitialises the Trickle timer following the MRD value, as in Equation 4.9. Indeed, the aim is to reduce the number of exchanged DIO messages, stabilise the network, and thus tolerate the attack.

$$\text{Timer} = \begin{cases} \text{Reset to MRD} & \text{if MRD} < \text{Current-interval } I \\ \text{Do not reset} & \text{else.} \end{cases} \quad (4.9)$$

The pseudocode in Algorithm 1 summarises the proposed approach.

## 4.6 Approach Evaluation

### 4.6.1 Performance Metrics

As presented in the literature [96] [100][118], the DIS attack influences significantly the control overhead, especially the number of DIO messages and energy consumption. Hence, the performance metrics used to evaluate RPL-MRC are as follows.

- Control packer overhead: It is the total number of DIO messages transmitted during the simulation.

---

**Algorithm 1** M-DIS and Sybm Attacks Prevention (Tolerance)

---

**Require:** MRC, I (Current-interval)

Calculate Maximum Response Delay (MRD) using MRC as defined in Equation 4.8

**if** a node receives a Multicast DIS message **then**

**if** MRD < I **then**

        I=MRD (i.e., Reset I to MRD)

        Select t in [MRD/2, MRD] (i.e., Delay the response by a random amount of time in the range [MRD/2, MRD])

**if** the number of response from its neighbours reaches the threshold defined by the border router **then**

            Cancel the pre-programmed response

**else**

            Send a Multicast DIO once the delay has expired

**end if**

**end if**

**end if**

---

- Power consumption (mW): It is the average power consumed by all the nodes in the network during the simulation. The calculation of the power consumption for each node is done by adding up the energy consumed on CPU (listening state), LPM (low power idle state), RX (radio listen state), and TX (radio transmit state).

In addition to the control overhead and power consumption metrics, we evaluated the data packets overhead.

- Data packets overhead: It is the total number of data packets received by the border router during the simulation. We also recorded the number of duplicated data packets to highlight the instability of the network.

## 4.6.2 Simulation Settings

Using the Cooja-Contiki simulator, we simulated three topologies of 30 nodes each, where 29 sender nodes transmit their packets to one sink node (i.e., a multipoint-to-point traffic). We used Tmote sky nodes and the radio protocol UDGM (Unit Disk Graph Radio Medium) with distance loss as a link failure model as it provides a real-world emulation of the lossy links and shared media collision among RPL's nodes. Additionally, we used the CSMA/CA for the link layer and the ContikiMAC as the radio duty cycling (RDC) protocol. Because the sender nodes are lossy by nature, the reception ratio (RX) was set to 70%, whereas the transmission ratio (TX) for all nodes was set to 100%, which means a loss-free transmission. The transmission range was set to 70m and interference range to 80m.

We set up three main scenarios: (1) RPL without attack, (2) RPL under attack, and (3) RPL with RPL-MRC. We implemented sub-scenarios, where we varied the number of attackers. We also varied the attack's frequency and the MRC values. We run every simulation for 15 minutes. The nodes were distributed in an area of 300m x 250m. We used the RPL-collect package for packets generation, where each node sends one packet of 46 bytes every 60 seconds. For the performance metrics, we used radio messages and collect-view tools. Five runs were conducted for each scenario, and values were averaged. Besides, the proposed solution has been evaluated for the SybM attack defined in [29] and for low data packet rate. Table 4.8 summarises the parameters used for the simulations.

### 4.6.3 The M-DIS Attack Frequency Effect

To evaluate the effect of the attack frequency on the RPL performance, five malicious nodes were distributed uniformly in the network to ensure covering the vast majority of benign nodes and maximise the network's damage. MRC is set to 15 (which is equivalent to MRD equal to 32,768 seconds). We selected MRC=15 because it gives the best results as it can be seen in Section 4.6.5. The attack is triggered within intervals of 3, 6, 10, 15, and 30 seconds. For instance, attack frequency 3 means that each attacker sends a Multicast DIS message every 3 seconds, which means it sends 20 Multicast DIS messages per minute. In the case of attack frequency 30, each attacker sends a Multicast DIS message every 30 seconds, which means it sends 2 Multicast DIS messages per minute. Native RPL (RPL), RPL under M-DIS attack (RPL-DIS), and RPL under M-DIS attack with MRC countermeasure (RPL-MRC) were evaluated in terms of the metrics in Section 4.6.1.

#### 4.6.3.1 Control Overhead

Figure 4.16 shows the performance of the network in terms of DIO messages overhead following different attacking intervals. It is noticed that the number of DIO messages sent in the RPL-DIS scenario is very high compared to the native RPL and RPL-MRC regardless of the frequency of the attack. We can observe that in the RPL-MRC scenario, the DIO overhead has been decreased by 86%, 84%, 84%, 81%, and 79% for 3, 6, 10, 15, and 30 seconds intervals, when compared to the RPL-DIS scenario. Indeed, RPL-MRC has performed very well, reducing the overhead to almost the one generated in the native RPL. This is because RPL-MRC is executed every time a Multicast DIS message is received, even from legitimate nodes. We notice that in some cases (e.g., 10s and 30s intervals) the overhead is lower than with native RPL. This could be because the nodes did not reset their timers according to the rule in equation 4.9 (i.e., the current time is less than MRD), in addition to the

Table 4.8: Simulation Parameters

Parameter	Value
Simulator	Cooja-Contiki 3.0
Simulation time (mn)	15
Network area	300x250m <sup>2</sup>
Node type	Tmote Sky (telosB)
N of nodes	30 (1 destination and 29 senders)
N Malicious nodes	2, 5, 10
Attack frequency (s)	3, 6, 10, 15, 30
MRC Values	13, 14, 15, 16, 17
Transmission range	70m
Interference range	80m
TX, RX	100%, 70%
MAC	ContikiMAC
Link failure model	UDGM with Distance Loss
Traffic rate	1pkt per 60s per sender
Packet size	46 bytes

execution of RPL-MRC that reduces DIOs response to legitimate DIS Multicast.

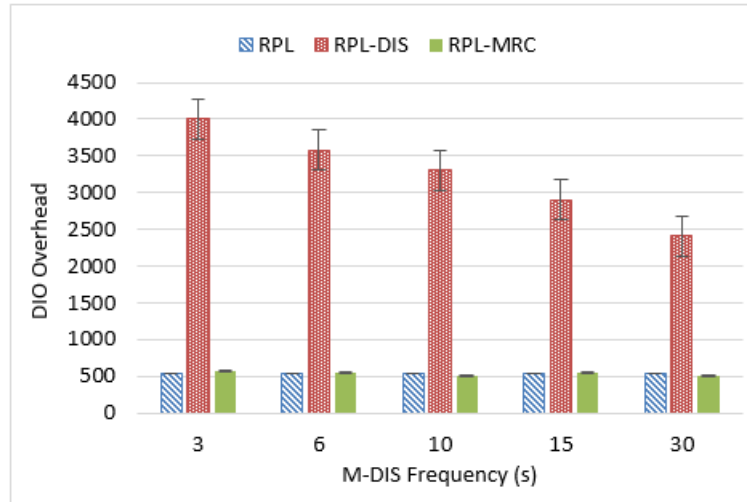


Figure 4.16: Control overhead vs attack frequency.

#### 4.6.3.2 Power Consumption

As observed in Figure 4.17, the RPL-DIS network suffers heavily in terms of power consumption due to the attackers being able to flood the network with many DIS and DIO messages. However, following the RPL-MRC mitigation mechanism, the average power consumption has been reduced by 53%, 48.5%, 48%, 41%, and 34% for attack frequency of 3, 6, 10, 15, and 30 seconds, respectively. Indeed, the decline in the number of transmitted DIOs has resulted in lower power consumption. Both



results (i.e., control overhead and power consumption) are justified by executing the RPL-MRC mechanism that redefines how to respond to a DIS Multicast.

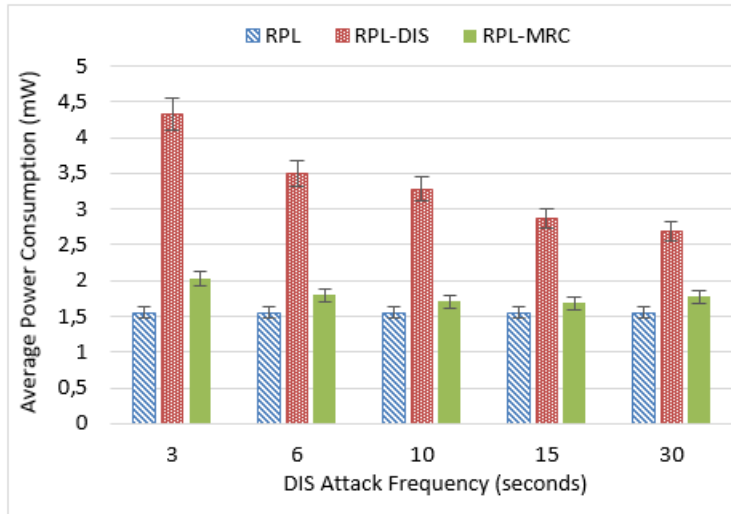


Figure 4.17: Power consumption vs attack frequency.

#### 4.6.3.3 Data Packets Overhead

The results in Figure 4.18 demonstrate that the data packets overhead increases under the M-DIS attack. This is due to the increase of DIO overhead, which suppresses communication channel availability, forms a locally unstable network, and thus induces generating duplicate data packets. We notice that both the number of duplicate data packets and the number of delivered packets have been reduced using RPL-MRC countermeasure. In fact, under RPL-MRC, the network is more stable because the DIO overhead is reduced significantly.

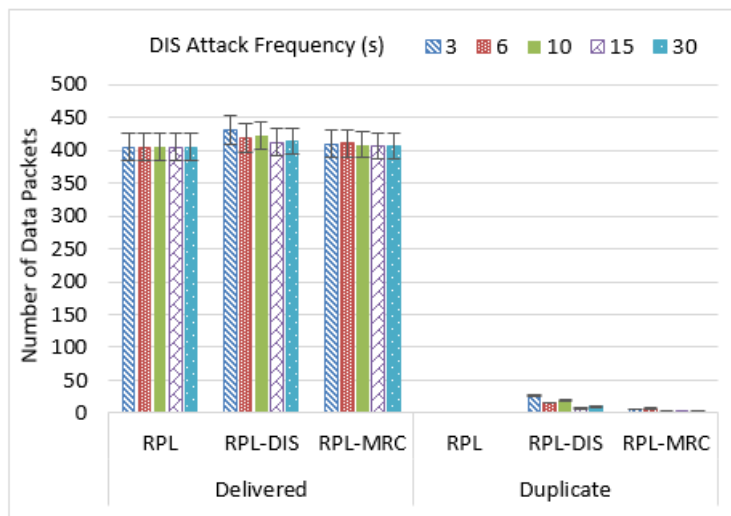


Figure 4.18: Delivered and duplicate data packets vs attack frequency.

#### 4.6.4 The Number of Attackers Effect

To evaluate how RPL-MRC performs according to the number of attackers present in the network, we implemented the M-DIS attack with different numbers of malicious nodes (2, 5, and 10), an attack frequency of 3 seconds and an MRC set to 15.

##### 4.6.4.1 Control Overhead

Figure 4.19 shows the impact of varying the number of malicious nodes on the amount of control message exchanged in the network. In the RPL-DIS scenario, the control overhead increases when the number of attackers increases because the attackers were distributed uniformly in the network. Thus, a large number of legitimate nodes are affected by the attack. In RPL-DIS, all nodes in a malicious node's radio range reset their Trickle timers every time they receive a DIS Multicast, and hence, send frequently DIO messages that are propagated in the network. However, RPL-MRC mechanism regulates the reset of the Trickle timer and the transmission of DIO messages in a way to reduce the overhead in the network. As a result, the overhead was decreased by 79%, 98.7%, and 90% in the presence of 2, 5, and 10 attackers, when compared to RPL-DIS.

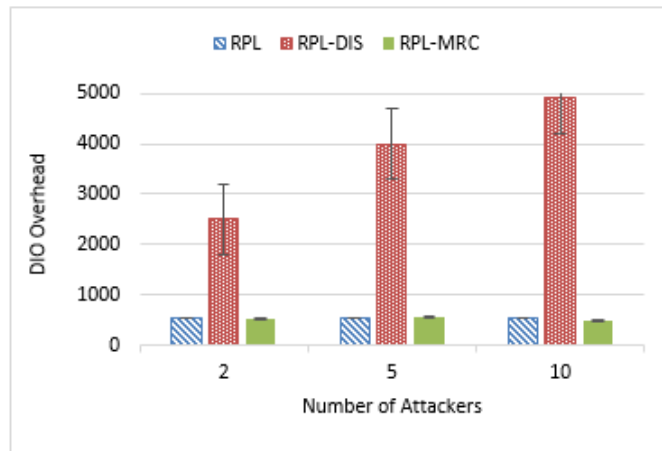


Figure 4.19: Control overhead vs number of attackers.

##### 4.6.4.2 Power Consumption

By analysing Figure 4.20, we realise that as the number of malicious nodes increases, the energy consumption increases significantly in the RPL-DIS scenario. Considering that more attackers exist in the network, more legitimate nodes respond to the attack by resetting their Trickle timers and sending more DIO messages, resulting in larger power consumption. Albeit the energy consumption under the RPL-MRC scenario is more than under the native network (the RPL scenario), it remains very

good compared to the network under attack (the RPL-DIS scenario). The RPL-MRC mechanism was able to decrease the control overhead for a different number of attackers and, consequently, the network's overall power consumption. Although the number of DIOs has decreased, we notice that the energy increases as the number of attackers increases. Indeed, malicious nodes consume more energy on transmitting DIS messages (frequency of 3 seconds per attacker), which means that the network's average power consumption increases.

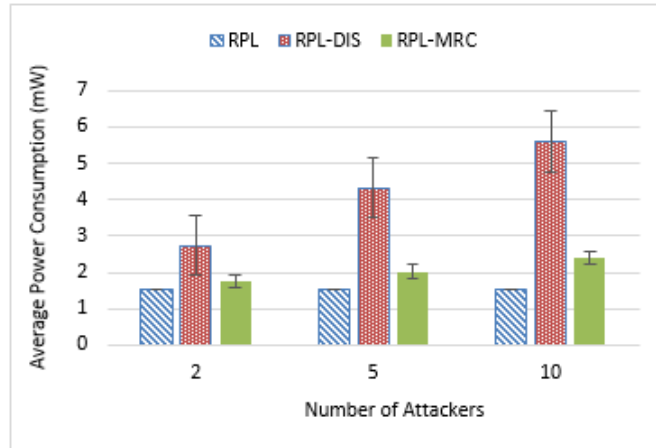


Figure 4.20: Power consumption vs number of attackers.

#### 4.6.4.3 Data Packets Overhead

Under M-DIS attack (for both RPL-DIS and RPL-MRC scenarios), the border router receives a larger number of original and duplicate data packets, as shown in Figure 4.21. When the node does not receive the acknowledgement, it schedules retransmission, leading to a duplicate packet. It is evident that data packets may be correctly received, and the corresponding acknowledgement may be lost or even may collide due to transmissions unreliability resulted from the increase of control overhead. However, regardless of the number of malicious nodes, RPL-MRC makes the network more stable. As a result, the number of duplicate packets is reduced.

#### 4.6.5 The MRC Parameter Effect

This section investigates the MRC parameter setting's effect on the RPL network performance by increasing the MRC value, starting with 13 and incrementing it by one to a maximum of 17 (i.e., 13, 14, 15, 16, and 17). The MRC values correspond to MRD values of  $2^{13}$ ,  $2^{14}$ ,  $2^{15}$ ,  $2^{16}$ , and  $2^{17}$ , respectively.

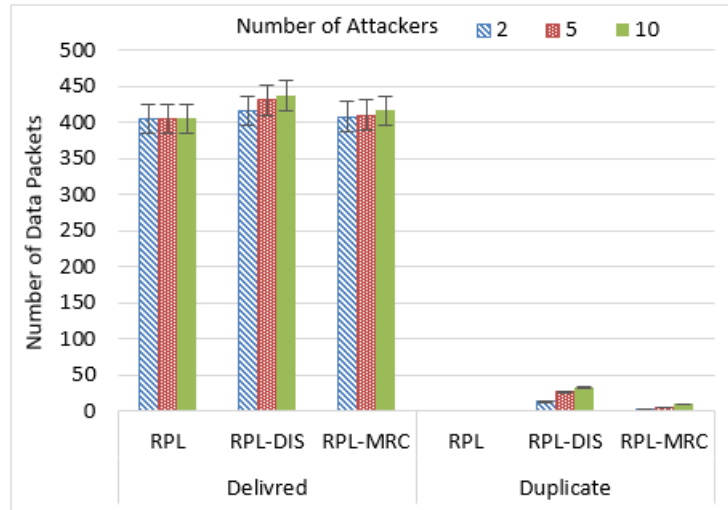


Figure 4.21: Delivered and duplicate data packets vs number of attackers.

#### 4.6.5.1 Control Overhead

It is clear from Figure 4.22 that RPL-MRC reduces the control overhead significantly, whatever the MRC value. Indeed, the control overhead has been decreased by 56%, 74%, and 86% for MRC equal to 13, 14, and 15, respectively. We notice that setting a small value for MRC getting closer (approximates) to the Trickle timer minimum interval (i.e., 13 and 14) induces more control overhead. Whereas, MRC values from 15 give approximately the same results and an overhead close to the native RPL one.

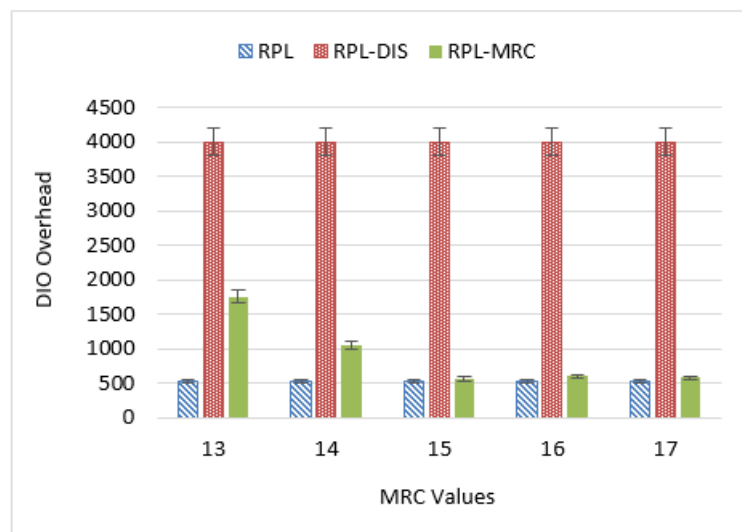


Figure 4.22: Number of DIO under different MRC values.

#### 4.6.5.2 Power Consumption

Similarly to all of the above cases, the decrease in power consumption under RPL-MRC is a logical consequence of reducing control messages overload. It is evident from Figure 4.23 that the MRC values from 15 give better results in terms of energy consumption. The following points can explain the results for both control messages overhead and energy consumption:

- The nodes reset their timers but suppress the delayed DIOs because the threshold of transmissions is reached.
- The nodes do not reset their timers because the current interval (period) is less than the response delay value. It could occur for MRC values of 15, 16, and 17.
- The nodes reset their timer to a value greater than the  $I_{\min}$  defined by the Trickle timer.

As in Section 4.6.4, malicious nodes consume more energy on transmitting DIS messages, which implies an increase in the overall network's average power consumption. However, the results remain satisfactory with a decrease between 35% and 53% compared to RPL-DIS.

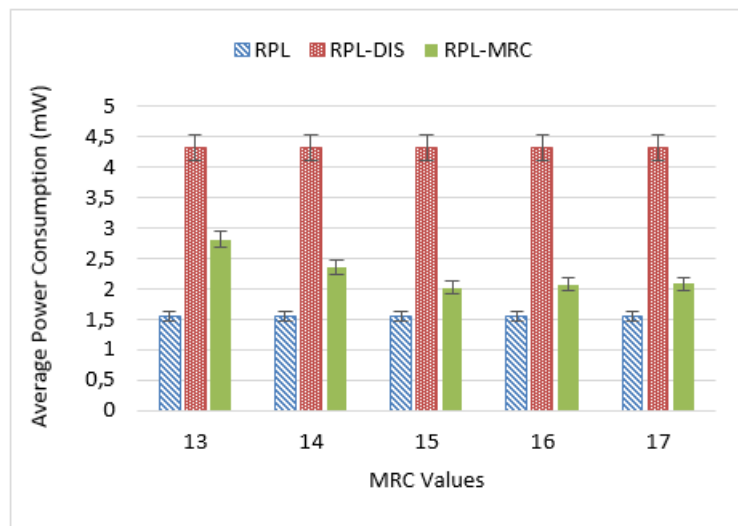


Figure 4.23: Power consumption under different MRC values.

#### 4.6.5.3 Data Packets Overhead

The results from Figure 4.24 have also demonstrated that the DIS attack may moderately affect data packets' delivery. RPL-MRC has improved the network's stability for all MRC values and consequently decreased the data packets overhead.

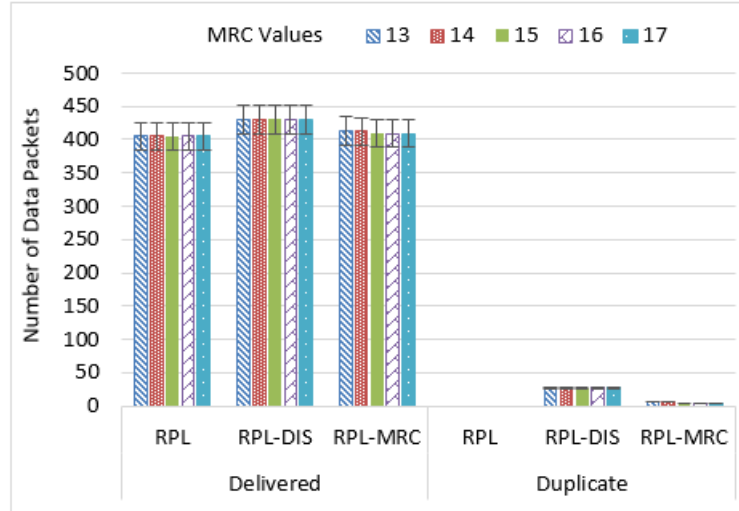


Figure 4.24: Delivered and duplicate data packets for different MRC values.

### 4.6.6 The Data Packet Rate Effect

In this section, we investigate the effect of RPL-MRC on RPL under the M-DIS attack by increasing the data packet rate to 1 packet every 10 seconds (i.e., 6 packets per minute, which results in approximately  $6 \times 15 \times 29$  packets generated for the whole simulation time). We set the number of attackers to 5, the attack frequency to 3 seconds, and MRC to 15.

#### 4.6.6.1 Control Overhead

Figure 4.25 demonstrates that the control overhead increases with the increase of the data packets sent during the simulation for the three scenarios. However, RPL-MRC reduces the control overhead significantly regardless of the data rate.

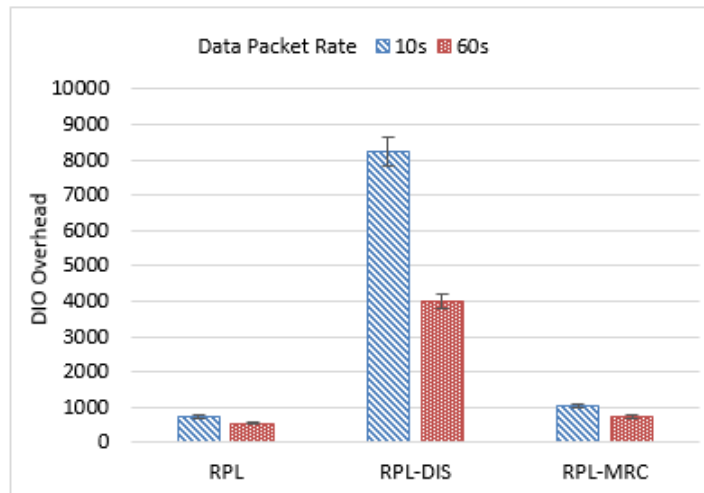


Figure 4.25: Number of DIO vs data packet rate.

#### 4.6.6.2 Power Consumption

From Figure 4.26, we notice that the average power consumption increases for all scenarios following the data packet rate. This is because the nodes consume more energy on transmitting both data and control packets. RPL-MRC mechanism was able to decrease the average power consumption as a consequence of decreasing the control overhead even in the case of an increased data packet rate.

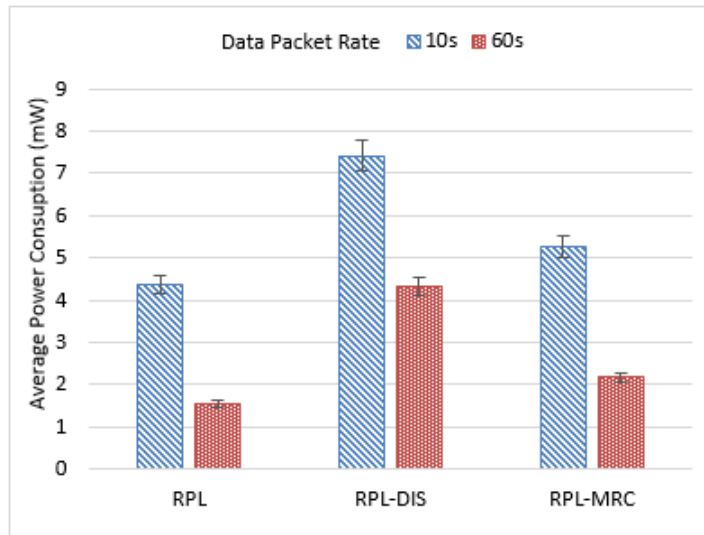


Figure 4.26: Power consumption vs data packet rate.

#### 4.6.6.3 Data Packets Overhead

As it can be seen in Figure 4.27, the increase in the packet rate affects the packet loss in all scenarios, especially in the presence of attackers (RPL-DIS and RPL-MRC). Nonetheless, the RPL-MRC approach reduces significantly the packets loss and duplication as it enhances the network's stability compared to RPL-DIS. As a conclusion, RPL-MRC is efficient even in the case of high packet rate.

## 4.7 Approach Evaluation under Mobility: SybM Case

As illustrated in Figure 4.2, SybM attack is a combination of Sybil and M-DIS attacks where malicious nodes are mobile. In this section, we study the effect of the proposed solution (RPL-MRC) on RPL under SybM attack. We simulated a network of 50 TelosB nodes (Sky motes) with one border router and 49 senders. Table 4.9 highlights the simulation parameters specific for SybM attack.

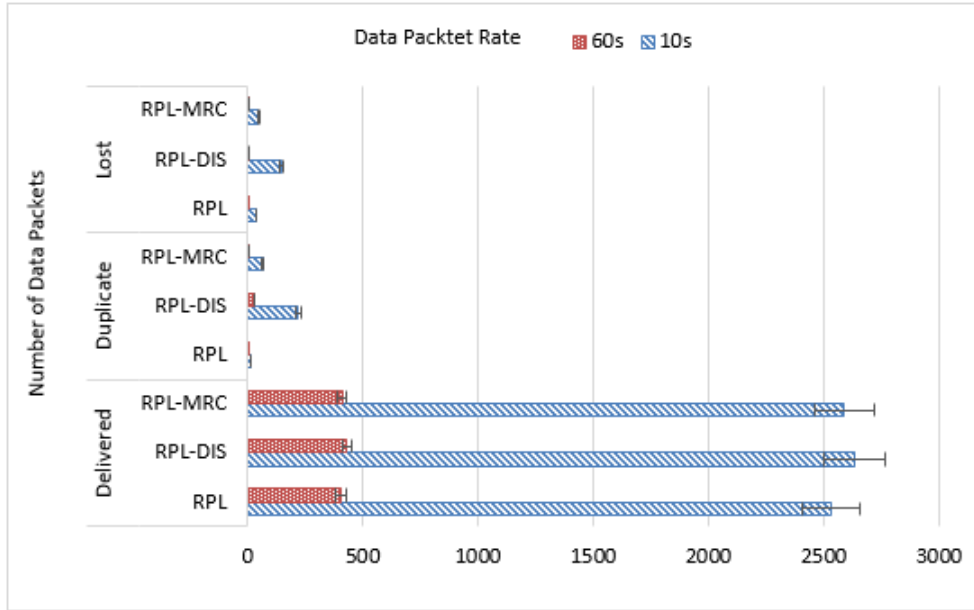


Figure 4.27: Delivered, duplicate and lost data packets vs data packet rate.

Table 4.9: Simulation Parameters for RPL-MRC under SybM Attack

Parameter	Value
Simulation time (s)	330
Network area	300x200m2
N of nodes	50 (1 destination, 49 senders)
N of malicious nodes	10
Attack frequency (s)	3
N of moves (identities)	5 per attacker
MRC Value	15

### 4.7.1 Control Overhead

Figure 4.28 demonstrates that the overhead generated in all scenarios exceeds the one generated in the previous sections, even if the simulation duration is 5 minutes. Actually, we used a larger network of 50 nodes that generate more traffic to construct and maintain the RPL topology. As depicted in the figure, SybM attack caused an extra overhead of 55.7%, which is more than the double compared to the one generated from native RPL. Nonetheless, RPL-MRC behaves like in the static case (RPL-DIS) and reduces the attack's effect (RPL-SybM) on control overhead by 55%. In conclusion, RPL-MRC is very efficient to reduce the response to a DIS Multicast in a dynamic (mobile) network.



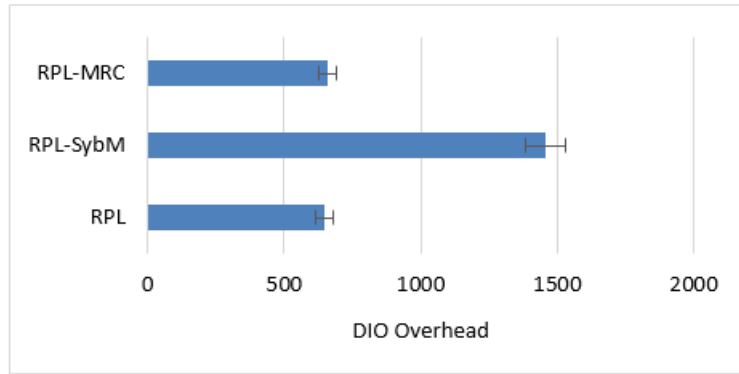


Figure 4.28: Control overhead under SybM attack.

### 4.7.2 Power Consumption

The power consumption increases with the size of the network following the increase in control overhead. The Figure 4.17, Figure 4.20, Figure 4.23, and Figure 4.29 reveal that the power consumption for native RPL with 50 nodes and 5 minutes simulation time increased by 21% compared to 30 nodes and 15 minutes simulation time. RPL-SybM generated an extra power consumption of 42%, and RPL-MRC reduced it by 33.6%. Hence, RPL-MRC additional overhead is about 8.4%, which is acceptable as the first line of defence.

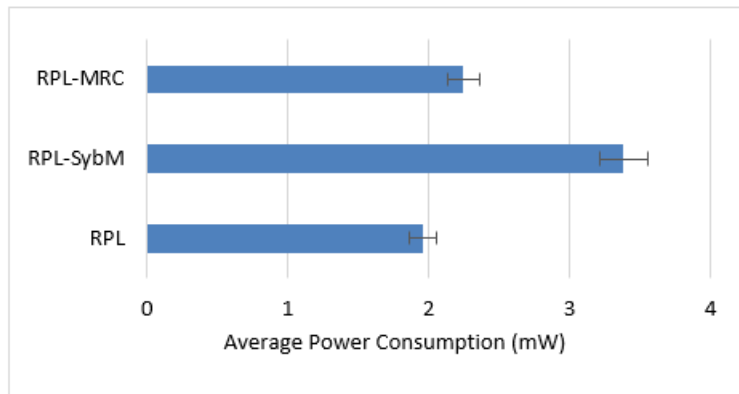


Figure 4.29: Power consumption under SybM attack.

### 4.7.3 Data Packets Overhead

As shown in Figure 4.30, in the presence of SybM attack, the duplicate data packets increase, hence increasing the delivered ones. This can be explained by the mobility of the attackers and the Multicast of DIS messages that render the network unstable. RPL-MRC succeeds in reducing the number of duplicate packets by 91%.

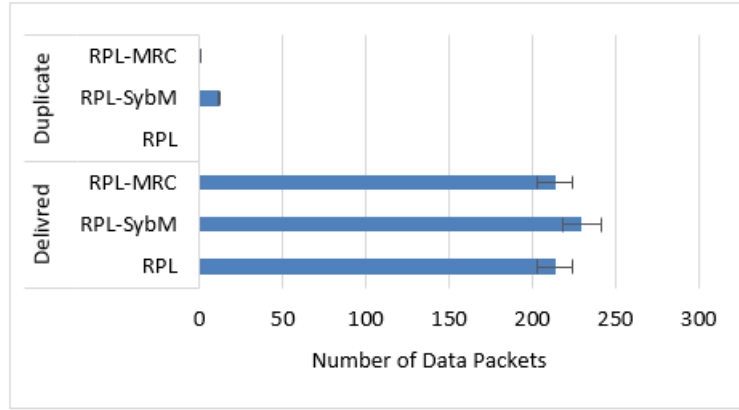


Figure 4.30: Data packets overhead under SybM attack.

## 4.8 Summary

In this chapter, we demonstrated that a simple DIS Multicast can significantly increase the number of exchanged control messages. The abrupt increase of control overhead increases the overall power consumption of the network and further reduces the network lifetime. Besides, we introduced SybM, a novel attack against RPL that combines the Multicats DIS attack, the Sybil attack and the node mobility. We presented an evaluation of the RPL's performance under both attacks. We introduced a solution to tolerate M-DIS and SybM intrusions. The results highlighted the efficiency of the proposed RPL-MRC mechanism for reducing control overhead, power consumption, and data packet overhead. We studied the effect of the approach for different scenarios (e.g., varying the attack frequency, varying the number of attackers, varying the proposed parameter MRC, varying the data rate, and under mobility). We conclude that RPL-MRC achieves high performance in all cases. We demonstrated the RPL-MRC scalability as it presents good fulfilment for a larger network (i.e., case of SybM attack). RPL-MRC can reduce the M-DIS and SybM attacks' effect before the attackers are detected and discarded from the network. We suggest that our solution could be combined with IDSs such as the specification-based or the anomaly-based to protect RPL-LLNs.

In the next chapter, we introduce two IDSs that can be used to detect intrusions against RPL networks.

# Chapter 5

## Intrusion Detection and Tolerance Systems for RPL's Security

Successful deployment of Low power and Lossy Networks (LLNs) requires self-organising, self-configuring, and security support. In this chapter, we introduce two IDS solutions to detect attacks against RPL networks. The first approach, named T-IDS is a specification-based IDS. T-IDS is a cross-layer trust-based IDS that copes with the RPL's mobility and identity issues. The second approach, named RF-IDS, is an anomaly-machine-learning-based intrusion detection and tolerance system that uses the Random Forests classifier to detect attacks and other mechanisms to tolerate attacks.

### 5.1 A Trust-based Intrusion Detection System for Mobile RPL Based Networks

Each IDS proposed in the literature has its advantages and disadvantages. The main weakness of the majority of them is the lack of mobility and secure identity handling that can be exploited by Sybil attackers [28][29].

Indeed, the impacts caused by RPL attacks and especially the Sybil ones require developing new mitigating mechanisms. Different approaches have been proposed to address the Sybil attacks issue [142]. However, these solutions are not desirable for several reasons. Some of the proposed solutions are energy costly, or limited to some types of networks (i.e., Sensor Networks or Ad hoc Networks), or primarily designed for non-mobile nodes. In the context of IoT, other approaches have been proposed in [143]. What makes Sybil attack more difficult to detect by existing approaches is the fact that malicious nodes intend to use one of their identities (i.e., IP addresses) at a time in one location. Hence, one Sybil identity is seen as one legitimate physical node. To overcome this type of attack, we propose a distributed, cooperative and

hierarchical trust-based IDS architecture that integrates three cooperative modules: IdentityMod, MobilityMod and IDSMoD as illustrated in Figure 5.1.

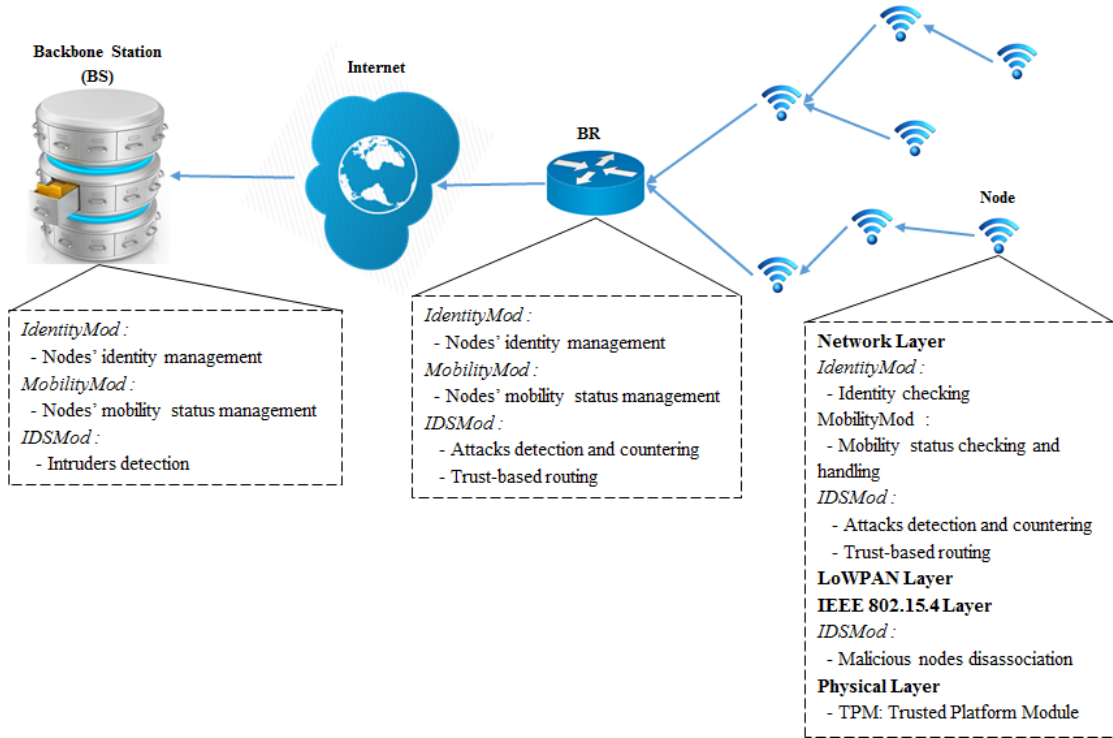


Figure 5.1: T-IDS architecture.

T-IDS is a hybrid-IDS because both the BR and in-network nodes collaborate in defending against internal attackers. Furthermore, T-IDS is trust-based for two reasons. First, a Trusted Platform Module is integrated to each in-network node. Second, nodes rely on a new collaborative trust metric evaluation when routing [114][115][116]. In the following sections we introduce the hybrid trust-based IDS actors and components and demonstrate how they can be used.

### 5.1.1 T-IDS Characteristics

1. RPL is based on the IPv6 Neighbor Discovery mechanism. It relies on multicast operations to setup the network topology. As discussed in 4, a simple multicast DIS message can affect the whole network. The problem associated with multicast NS (Neighbor Solicitation) and NA (Neighbor Advertisement) messages are more frequent in large-scale radio environments with mobile devices, which exhibit intermittent access patterns and short-lived IPv6 addresses [150]. The works proposed in [150] enables to lower the rate of RA (Router Advertisement) messages by extending the Address Registration Option (ARO), but does not solve the multicast associated problems. In T-IDS, RPL itself will be adapted to reduce the response to multicast messages in the case of mobile nodes as discussed in Chapter 4, Section 4.5.

- RPL relies on IPv6 addresses to identify nodes within the network. The same node can change its IPv6 address (i.e., Sybil attack) and try to join the network using the new address as a new identity. In T-IDS we propose a centralised beforehand registration of nodes. Each node has an associated unique identifier, which will be conveyed within the control messages along with its IPv6 address (see Figure 5.2). The identifier will be used by T-IDS's modules to detect and report intruders. This is inline with the IETF approach to introduce registrars [150].

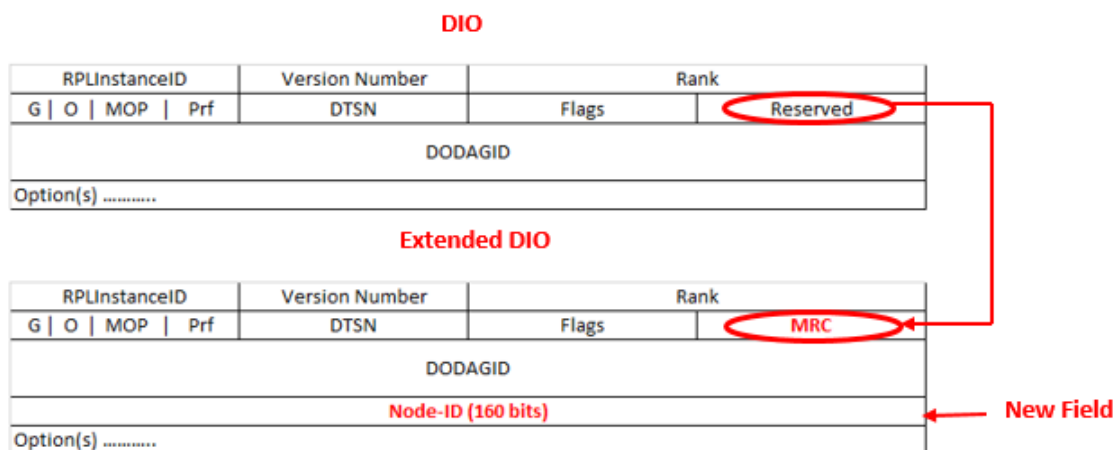


Figure 5.2: New DIO message format.

- In the trust-based RPL scheme [114][115][116], the in-network nodes collaborate to detect intruders using a trust-based routing scheme. In T-IDS, a mitigation method is induced as a third line of defence. The IDS reacts in a corrective action. This is done by executing trust-based RPL where nodes avoid malicious and suspicious nodes when selecting their routing paths. In T-IDS, trust calculation is enhanced by adding a new trust component; Mobility.

### 5.1.2 T-IDS Actors

T-IDS is composed of a centralised Backbone Station (BS) that federates multiple 6LoWPAN/LLN sub-networks. The BS may be part of anycast group for redundancy issue. Each 6LoWPAN/LLN sub-network is attached to the BS via a Border Router (BR). BRs are responsible of monitoring the in-network nodes and make the global intrusion detection decisions by associating and aggregating intrusion alerts from in-network nodes. Each in-network node monitors in a trusted-collaborative way its neighbours to detect intrusions. The BS and the BR are both supposed to be trusted entities. Figure 5.3, Figure 5.4, and Figure 5.5 depict BS, BR and in-network nodes operations, respectively.

### 5.1.2.1 Backbone Station (BS)

maintains the list of all Network Nodes (NNs) and their respective states. The BS handles the list of nodes authorised to access the network. In NNs, to each node is associated a TPM-ID, a Node-ID associated to the TPM-ID, the Node-Status flag (i.e., Mobile, Static), and the BR prefix associated to the node after deployment. When a node wants to join the network, it must be first registered at the NNs list. In addition, the BS maintains a list of potential MALicious Nodes (MAN) for all BR sub-networks.

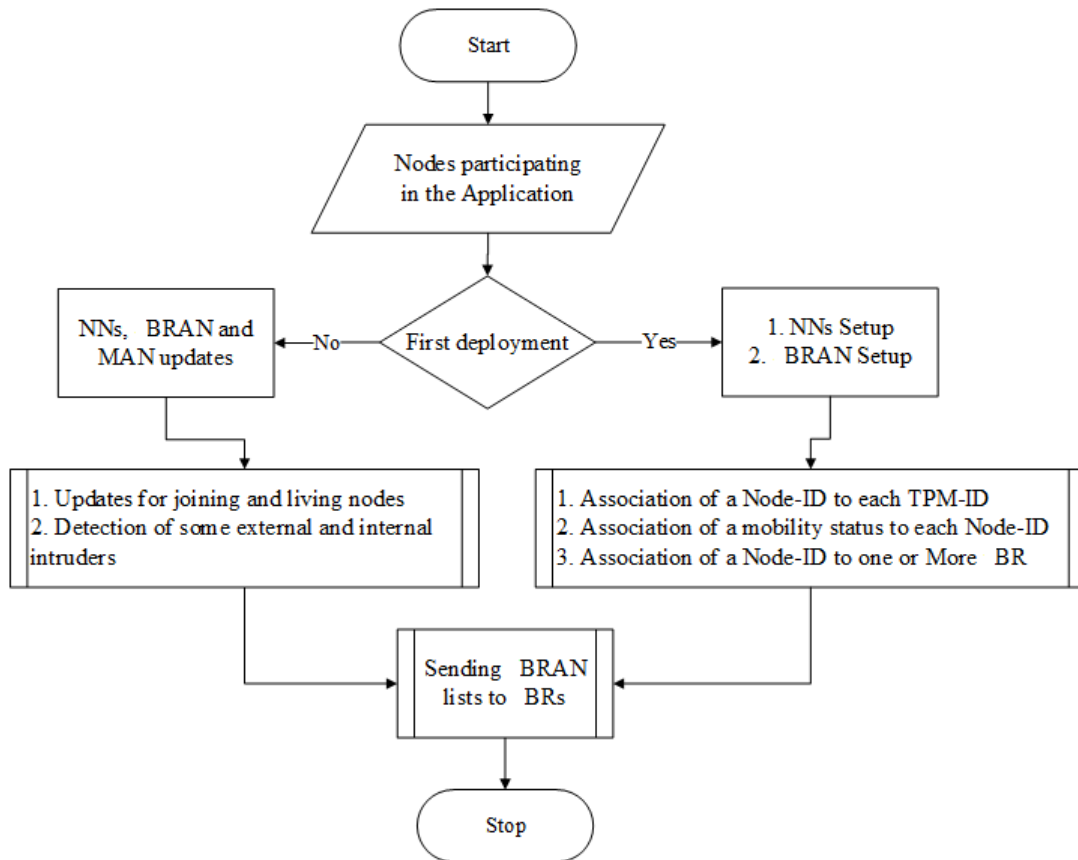


Figure 5.3: The backbone station operations.

### 5.1.2.2 6LoWPAN/LLN Border Router (BR)

maintains three dynamic lists: the first list contains BR Area Nodes (BRAN) within the BR's IPv6 prefix. BRAN is elaborated and updated by the BS and transferred to the BR using a secure channel. The second one contains MOBILE Nodes (MON) and the third list contains the MALicious Nodes (MAN). The BR is responsible for setting the MRC field in the DIO message as presented in Chapter 4, Section 4.5.

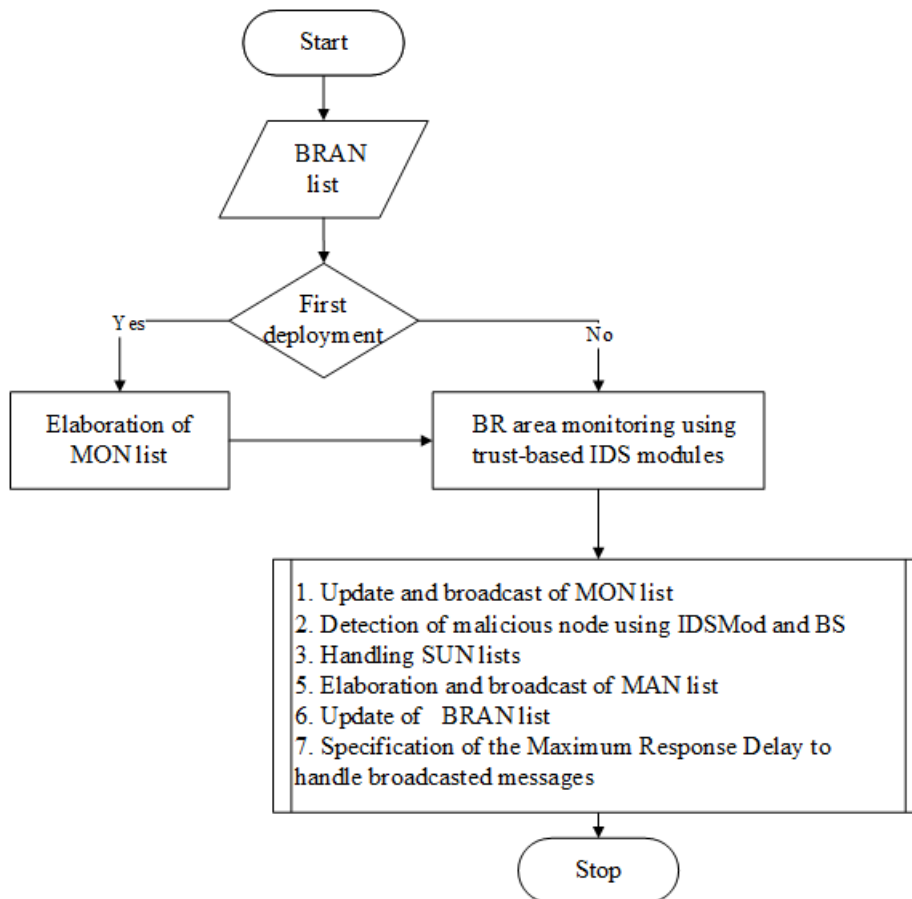


Figure 5.4: The border router operations.

### 5.1.2.3 Monitoring Nodes (MNs)

each in-network node is a MN by default. MNs maintain a list of SUSPICIOUS Nodes (SUN) and a list of malicious nodes (MAN). They also keep a copy of MON list elaborated by the BR. The lists are stored in the TPM. It is assumed that a node is already registered with one BR in the BRAN list.

## 5.1.3 T-IDS Modules

### 5.1.3.1 Module for identity management (IdentityMod)

The Identity Module (IdentityMod) is used to control access to the network. Each node, which is part of the network or try to join the network must have a unique identity to limit exposure of the network to attacks from unauthorised nodes. To the handle identity issue and off-load security feature, each node is equipped with a Trusted Platform Module (TPM), which provides uniquely unforgeable identity for the node (TPM-ID). TPM is a cryptographic co-processor chip known to be used in building hardware support identification, storing security parameters, and handling cryptography calculation. In our approach, manufacturers are required to

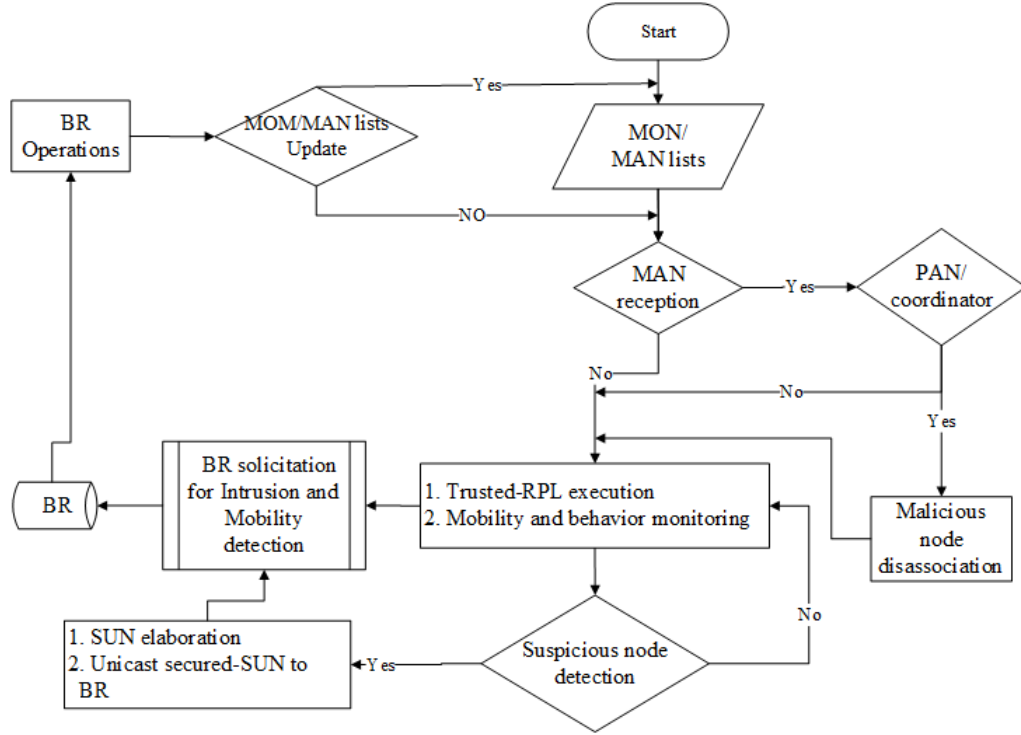


Figure 5.5: In-network nodes operations.

equip each device with a TPM chip before factory. One component of a TPM is the Endorsement Key (EK); a public-private RSA key pair created during manufacture. The public EK value will not change during the TPM's lifetime and it represents the TPM-ID (i.e., Node-ID). Besides EK, each node within the network uses two different symmetric keys: a Long Term Key (LSK) shared with the BR, and a Group-Key (GK) shared between all nodes. All symmetric keys are stored in the TPM chip. GK will be used to secure RPL's control messages. If an insider attacker compromises one node it gains access to the GK, and the security of the whole network is compromised. As a consequence, LSK will be used to send securely data packets and security related messages to the BR. The authenticity and integrity of exchanged messages between the BR and a particular node can be secured using lightweight IPsec with LSK [55].

After nodes' deployment, and before starting the construction of the RPL topology, the BS uses IdentityMod to set-up the BRAN list for each BR within the network. This list will be used to control access and authenticate nodes. To authenticate a node at any stage of the network execution, RPL control messages should convey besides the IPv6 address of the node, its unique identifier. In other word, the identifier of each node has to be embedded in the 6LoWPAN packets. In addition, each node records the identifier associated to the IPv6 address in its routing table. In this way, even if the nodes autonomously calculate their IP addresses while moving, they could be authenticated using their identifiers Node-IDs. Once an attack is



detected, the responsible nodes will be known. If a node detects that another node is malicious, it updates the SUN list with the identity of the suspicious node and sends it securely to the BR using LSK.

The BS uses IdentityMod to associate to each TPM-ID 20 bytes long Node-ID. Thus, the Node-ID is a cryptography-based unique representation of a node derived from the TPM-ID. For RPL networks, the MAC Maximum Transmission Unit (MTU) size is about 127 bytes. And the size of the TPM-ID varies from 64 to 254 bytes depending on manufacturing. To handle control overhead issue caused by a large number of fragments for the same message, we propose to shorten the size of the node's identifier from 64-254 bytes to 20 bytes long using the SHA1 hash function. In the proposed solution, we extend DIO, DIS and DAO control messages with 20 Bytes before Options Object to carry Node-ID [22]. Figure 5.2 depicts how DIO is extended. DIS and DAO are extended in the same way.

In [151], the authors proposed to extend the address registration option (ARO) for 6LoWPAN ND with a cryptographic identifier field. However, nodes compute themselves a cryptographically unique identifier and associate it with one or more of their IPv6 registered addresses, which leads to computation overhead. In addition, their solution is mainly designed to handle addresses duplication, and thus can be used to mitigate CloneID attacks.

### 5.1.3.2 Module for mobility management (MobilityMod)

Mobility is also handled according to a hierarchical manner with the collaboration of the BS, BR and in-network nodes. MobilityMod is used by the different actors to maintain the state of the network regarding mobile nodes. Indeed, BRAN contains the mobility status of each node. Upon receiving BRAN from the BS, the BR defines a new list by keeping only the mobile nodes; MON: MOBILE Nodes. After the construction of RPL, the BS broadcasts MON to all nodes. Hence, mobile nodes are known by all in-network nodes, and thus using its identity (Node-ID) the presence of the mobile node is determined by the neighbouring nodes. In other words, when a node constructs its routing table, it uses the MON list to check and monitor the mobility status of each neighbour. From this point, if any moving node sends a DIS message using a new IPv6 address, its neighbours can detect it as a suspicious node (i.e., same Node-ID with a different IPv6 address) and add it to their respective SUN lists. Furthermore, if any moving node sends a DIS message using a new Node-ID and a new IPv6 address, its neighbours can check the MON list. If the node does not exist on the MON list, it will be detected as suspicious (i.e., node not registered within BRAN) and add it to SUN lists. In addition to MON list, and to handle mobility, each node verifies the RSSI (Received Signal Strength Indication) of its respective neighbours. If the RSSI value of a monitored node has

degraded or has been null, this could be due to the fact that it is a malicious mobile node that has not been added in the MON list. In all cases, the monitoring node considers that node as suspicious, updates SUN list and unicast it to the BR using LSK.

If a mobile node sends packets with an identifier not known by the BR (i.e., not present in the BRAN list); the BR sends a request to the BS to ask if the new mobile node belongs to the network. If the mobile node is a legitimate node, the BS replies by sending an updated BRAN list containing the identity of this node. Nevertheless, if the node is not previously registered in the NNs list, the BS informs the BR that the node is an intruder. If there is any node that joins or leaves the BR's network, the BR will update the MON list, and triggers a global repair with the new MON list. In the same way, if there are any intruders, the BR will update the MAN list and broadcast it to its 6LoWPAN/LLN's nodes.

MobilityMod can be used to obtain the localisation of the mobile malicious node in the network. This can be done by gathering mobility information from the neighbour list (i.e., routing table) of different static nodes.

### 5.1.3.3 Module for intrusion detection (IDSMod)

To detect attacks, each time the IDSMod will query the IdentityMod and the MobilityMod to verify if the node belongs to the network and if it is a mobile node. From the one hand, there is no mechanism in RPL for the nodes to monitor the behaviour of their neighbours. From the other side, attackers generally focus on specific behaviours and repeat them in high or low rates. Consequently, with minimal knowledge, and by observing and collaborating, nodes can detect misbehaving nodes. In this context, we propose to consider some appending for RPL to be used in IDSMod:

1. The first one consists on the integration of the new trust-based RPL scheme (named MRTS as Metric-based RPL Trustworthiness Scheme) proposed in our previous work for attacks countering [114][115][116]. In T-IDS, an enhancement of MRTS is proposed to detect misbehaving nodes in a dynamic environment. In MRTS, nodes within the network collaborate to detect malicious nodes according to specification-based behaviours. Periodically, each node calculates trust values of its one hop neighbours;  $\text{Trust}_{ij}(t)$ . Moreover, the node receives trust values evaluations of other nodes from its neighbours and aggregates all received and calculated trust values. The final trust values represent the result of collaboration of different participating nodes. In IDSMod, if a trust value of a node is less than a threshold, the node identity will be added to the SUN list, the list will be encrypted (using LSK), and sent

in unicast to the BR. Upon receiving SUN lists, the BR processes them and creates a new list containing malicious nodes; MAN. MAN list will be then broadcast to all nodes. MRTS uses four components to assess each node trustworthiness; honesty, unselfishness, energy and ETX. In the IDSMoD solution, we propose to use a new trust component namely mobility when calculating trust values as in Equation 5.1.

$$\left\{ \begin{array}{l} Trust_{ij}(t) = w_1 Trust_{ij}^{honesty}(t) \\ \quad + w_2 Trust_{ij}^{energy}(t) \\ \quad + w_3 Trust_{ij}^{mobility}(t) \\ w_1 + w_2 + w_3 = 1 \end{array} \right. \quad (5.1)$$

Where *Trust* represents the trust value evaluation of the node *i* for its neighbour *j* at time *t*, and takes values between 0 and 1.  $w_1$ ,  $w_2$  and  $w_3$  are weights associated respectively to the three trust components: honesty, energy and mobility.  $Trust_{ij}^{honesty}(t)$  is calculated by IDSMoD, where the node is evaluated as malicious or not according to its behaviour.  $Trust_{ij}^{mobility}(t)$  is calculated by MobilityMod using MON list and RSSI. In a very dynamic environment, the weight of the mobility component ( $w_3$ ) can have the biggest value.

2. The second appending consists of dealing with security related multicast messages such as the DIS and SybM attacks, as introduced in Chapter 4. This solution can be extended to be used for different kinds of multicast messages within the RPL network. Each of which may require its own delayed response. Thereby, control overhead can be reduced especially in the presence of an attacker.
3. The third appending consists on introducing a cross-layer scheme, where information collected from the network layer is used to discard malicious nodes from the link layer. Because IDSMoD is a cross-layer based IDS, if a suspicious node is set as malicious by the BS or the BR, the BR will broadcast MAN list to the whole network. Upon receiving MAN by WPAN (Wireless PAN) Coordinator associating the malicious node, the coordinator sends a disassociation notification to remove the malicious node from the WPAN. As a result, the malicious node will be totally isolated from participating in the network operations.

Our proposed scheme can deal with SybM attack as depicted in Algorithm 2.

---

**Algorithm 2** SybM tolerance, detection and countering

---

**Require:** MON, MAN, SUN

Upon receiving a multicast DIS message

**Step 1 (Tolerating the intrusion):** The receiving node delays responding to the message according to the RPL-MRC approach defined in Chapter 4, Section 4.5

**Step 2 (Detecting and countering the intrusion):** Meanwhile, receiving node uses the Node-ID field in the DIS message and queries IdentityMod and MobilityMod to verify if the node belongs to the network (using the routing table and/or querying the BR), and if it is a mobile node (checking if the sender is in MON list)

**if** Node-ID $\in$ MON **then**

Evaluate Node-ID trust value (Trust) in collaboration with neighbouring nodes

**if** Trust < Threshold **then**

1. Add Node-ID to SUN list
2. Send encrypted SUN list to the BR
3. Execute trust-based RPL routing by avoiding Node-ID

**else**

1. Querying BR and Waiting for a  $\delta$  time
2. If BRAN not yet updated, BR query the BS

**if** Newly deployed mobile node **then**

1. BR updates MON and Broadcasts it to its 6LoWPAN/LLN area
2. Upon receiving MON, in-network nodes update RPL routing

**end if**

**if** Newly deployed static node **then**

If not receiving MON or MAN by BR after the  $\delta$  time, update RPL routing

**end if**

**if** Malicious node **then**

1. Add Node-ID to MAN list by the BR
2. Broadcast MAN list by the BR
3. Upon receiving MAN list:

**if** malicious Node-ID associated WPAN Coordinator **then**

Store MAN and Send a disassociation request to discard the malicious node

**else**

Store MAN list

**end if**

**end if**

**end if**

**end if**

---

### 5.1.4 T-IDS Advantages and Limitations

In T-IDS, in the presence of a unique identification mechanism handled by a single entity (BS), the network can be protected easily from outsider attackers. In addition, the ability of the network to react by self-organising and working properly in the presence of attackers, using the enhanced trust-based RPL scheme by identity and mobility management, allows it to be protected from both insider mobile and static attackers. In term of computing and storage, T-IDS has the advantage of off-loading security-related computations and storage into TPM co-processor of each node. In fact, using TPM allows T-IDS to have greater processing power to implement strong security scheme. In addition, the generation of Node-ID is done by the BS, which is a powerful and trusted entity. However, we believe that an additional storage cost is needed for node's routing table to keep track of Node-ID in RPL routing processes. Furthermore, an additional communication overhead can come from the extension of control messages with Node-ID field, which leads to more fragmentation. Moreover, the size of DIO messages are likely to be more important when executing the collaborative RPL construction using the new trust metric [114][115][116]. Nevertheless, there is a need to a trade-off between strong security mechanism and extra overhead. Besides, T-IDS is a specification-based IDS that inherits the disadvantages of this type of IDSs (see Chapter 3-Section 3.8), and thus can only detect specific attacks.

In the next section, we introduce RF-IDS, a fault-tolerant artificial-intelligence-based IDS for RPL's security. Actually, ML-based IDSs are known for their high detection accuracy, especially for massive data volumes.

## 5.2 Fault-Tolerant AI-Driven Intrusion Detection System for the Internet of Things

Even though there are several methods to implement an IDS, artificial intelligence-based technologies, such as Machine Learning (ML) techniques are highly recommended. From the one hand, researchers are exploiting ML algorithms in IDS development because they are ideal for classification problems, especially with the good results that they achieve in the different domains. From the other hand, ML methods have an interesting potential in detecting unknown/zero day attacks that bypass traditional IDSs such as the signature-based and specification-based ones. Furthermore, an ML-based IDS employs statistical, genetic and heuristics or a combination of them to learn from previous experiences without explicit programming.

Therefore, ML can be applied at RPL nodes, fog/edge/BR nodes and (or) cloud nodes to extract and analyse from large-scale data, and hence detect malicious

behaviour. In this study, new RPL attacks based datasets are generated. Besides, different ML algorithms and a Deep Learning (DL) model are explored to develop an efficient IDS for RPL to detect and classify unseen routing attacks. A comprehensive evaluation of several experiments of these ML and DL classifiers are shown on the developed datasets.

### 5.2.1 RPL Intrusions

As presented in Chapter 3, various RPL attacks have been analysed in the literature, precisely, Rank attacks, Neighbour attack, DAO attacks, DIS attack, Version number attack, Local repair attack, HelloFlooding attacks, Selective forwarding attack, Sinkhole and Blackhole attacks, Wormhole attack, and Sybil and CloneID attacks [27]. Most of the efforts to secure RPL focused on detecting and/or countering Rank, HelloFlooding, Selective forwarding, Sinkhole, Blackhole, Version number, and Wormhole attacks as they represent the most harmful attacks [152]. In this contribution, we investigate the detection of the following six attacks: Decreased Rank (DR), Sinkhole (SH), Blackhole (BH), Selective Forwarding (SF), HelloFlooding (HF), and Version Number (VN).

### 5.2.2 ML Methods

In order to choose the best ML method for our needs, we evaluated the performance of various algorithms in a binary and multi-classification on the developed datasets. These classifiers are Decision Tree (DT), Random Forests (RF), K-Nearest-Neighbour (KNN), Logistic Regression (LR), Naive Bayes (NB), and Multi-Layer Perceptron (MLP) classifiers, in addition to a sequential Deep Learning (DL) model presented in Chapter 2-Section 2.5.

Regarding the parameters of the ML algorithms, we tried different configurations and chose the default parameters from the Python3's Scikit-learn library, as there are no big changes in results. We implemented the following classifiers from Scikit-learn:

- DecisionTreeClassifier,
- RandomForestClassifier with number of estimators (i.e., the number of trees in the forest) equal to 100,
- KNeighborsClassifier with k=10,
- Gaussian Naive Bayes classifier (GaussianNB),
- MLPClassifier with one hidden layer, 100 neurons, and the 'relu' activation function,

- and LogisticRegression (Logit) classifier with ‘sag’ solver.

For the DL model, we implemented a sequential Deep Neural Network (DNN) using the Python3’s Keras<sup>1</sup> library. DNN is an application of ANNs with multiple hidden layers, which uses backpropagation technique for training. The training is about feedforward and back-propagation phases. In the first phase, the hidden and output nodes calculate their activation functions. The second phase aims to propagate back the error, which is the difference between the output and the target value, from the output to the input. This step is about adjusting the different weights of the different neurons composing the network.

The implemented model includes 1 input layer, 5 hidden layers and 1 output layer. 50 neurons are used in the first and fifth layers, whereas 100 neurons are used in the second and fourth layers, and 300 neurons in the third layer. The DL model is the same as the one in [139]. We choose the model proposed in [139] to compare it with other ML models as the authors presented good results on RPL’s attacks detection.

### 5.2.3 Performance Evaluation Metrics

A clean and unambiguous way to present the prediction results of a classifier is to use a Confusion Matrix (CM). For a binary classification problem such as intrusion detection, the matrix has two rows and two columns as depicted in Table 5.1, where 0 and 1 are labels for normal and attack, respectively. Across the top are the predicted class labels and down the side are the observed class labels. Each cell contains the number of predictions made by the classifier that fall into that cell.

TP that represents the normal RPL traffic correctly classified as normal, TN that represents RPL attack samples correctly classified as intrusions, FN that represents RPL attack samples incorrectly classified as normal RPL traffic, and FP that represents normal RPL samples incorrectly classified as intrusions are used to determine the different metrics to assess the performance of a classifier [153]. In our study, we focus on accuracy, precision, recall and F1-score metrics to compare the classifiers presented in Section 2.5.

Table 5.1: Confusion matrix

	<b>Predicted 0</b>	<b>Predicted 1</b>
<b>Actual 0</b>	True Positive (TP)	False Negative (FN)
<b>Actual 1</b>	False Positive (FP)	True Negative (TN)

---

<sup>1</sup>Keras is an open-source neural-network library written in Python

### 5.2.3.1 Accuracy

The accuracy is defined as the ratio of correct predictions to the total number of all predictions, as in Equation 5.2.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (5.2)$$

### 5.2.3.2 Precision

The precision, also called the Positive Predictive Value (PPV), is the number of positive predictions divided by the total number of positive class values predictions, as in Equation 5.3.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5.3)$$

### 5.2.3.3 Recall

The recall, also called sensitivity, True Positive Rate (TPR), or Detection Rate (DR), is defined as the ratio of positive predictions to the number of positive class values in the test data, as in Equation 5.4. Recall is one of the most important metrics in the security context as it has the ability to calculate successfully detected intrusions.

$$\text{Recall} = \text{DR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5.4)$$

### 5.2.3.4 F1-Score

The f1-score, also called the F Score or the F Measure, is defined as the harmonic mean of precision and recall. It, thus, conveys the balance between the precision and the recall as in Equation 5.5.

$$\text{F1 - score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5.5)$$

## 5.2.4 Dataset Generation

There exist datasets available publicly that are commonly used for intrusion detection research such as KDDCUP 99, ISCX and NSL-KDD. Nevertheless, one gap in the field of IoT networks is the unavailability or privacy of developed RPL related datasets such as the IRAD dataset [139]. Indeed, building an RPL ML-based IDS requires one or multiple RPL-related datasets, where the model can learn from. The datasets shall represent normal and malicious RPL-based traffic. In our experiments, we elaborated two types of datasets: one dataset for each attack and one multi-class dataset for all the six attacks in Section 5.2.1.



#### 5.2.4.1 Simulations Setting

We used the simulator Cooja-Contiki for our experiments. Contiki is a powerful tool for building 6LoWPAN-IoT networks and has realistic results. Because the simulation of large topologies requires high memory and computing power, we deployed the simulator on a virtual machine with 48 GB RAM and 8 VCPUs on a server.

We simulated three topologies of 25, 50 and 100 TelosB nodes (Sky motes) with one root. Firstly without any attack, and then with 2, 4, and 10 malicious nodes, respectively. We implemented each attack separately. We used UDGM with distance loss as link failure propagation model as it provides a real-world emulation of the lossy links and shared media collision among RPL's nodes. The simulations duration is one hour with one packet of 46 bytes sent every 10 seconds. We used RPL-collect package for packets generation. Furthermore, we used the cooja-radio-logger-headless plug-in to capture traffic and generate PCAP files. We exploited the PCAP files to generate the datasets and extract features. Table 5.2 summarises the parameters used for the simulations.

Table 5.2: Simulation Parameters

Parameter	Value
Simulator	Cooja-Contiki 3.0
Simulation time	3600s (1 Hour)
MAC	ContikiMAC
Number of nodes	25, 50, 100
Number of malicious nodes	2, 4, 10
Transmission range	50m
Interference range	60m
TX, RX	100%, 90%
Network area	300x300m <sup>2</sup>
Propagation model	UDGM with Distance Loss
Traffic rate	One packet every 10 seconds
Packet size	46 bytes

#### 5.2.4.2 Feature Engineering and Selection

We implemented Python scripts for datasets' generation. We used Pandas, Numpy, and Scikit-learn libraries for features engineering, extraction and selection, Matplotlib and Seaborn libraries for data visualisation and plotting, and Scikit-learn and Keras libraries for data analysis.

**5.2.4.2.1 Features Extraction and Transformation.** We used Wireshark tool to transform the generated PCAP files to CSV files. The latter were pre-

processed using Python scripts. Initially, each CVS dataset includes six features: the packet sequence number ( $N^o$ ), simulation time (Time), source IPv6 address of the node (Source), destination IPv6 address of the node (Destination), the packet length (Length), and the packet information (Info).

Firstly, we simplified data in the CVS files such that nominal attributes are converted into discrete ones. For instance, source and destination IPv6 addresses were reduced to nodes' ID, and the broadcast address to the value 9999. In addition, the packet information DIS, DAO, DIO, Ack, and UDP was encoded 1, 2, 3, 4, and 5, respectively. To calculate datasets' feature values correctly, we first sorted the CVS files by simulation time. Then, we divided all the simulation time into periods of one-second duration (i.e., 1000 ms windows); to make better use of the extracted features for intrusion detection.

- In the DR and SH attacks, the malicious node advertises a Rank lower than the other nodes or a Rank equal to the BR's using a DIO message. When DR and SH attacks are triggered, normal nodes add the malicious node to their routing table and send their packets through it. Consequently, the number of received packets of the malicious node increases, as well as DIO and DAO counts. Accordingly, features such as Reception Rate, Reception Average Time, Received Packets Counts, Total Reception Time, DIO and DAO packets count should be added to the dataset.
- When VN attack is performed, the malicious node sends illegitimacy a DIO message with a new version number, thus triggering a global repair and pushing all nodes to exchange control messages. As a result, the DIO and DAO packets counts increase and should be used as features for the attack detection.
- When HF attack is performed, the malicious node sends illegitimacy DIS message pushing the neighbouring nodes to exchange control messages. As a result, the number of transmitted packets increases, as well as DIS packets count. Hence, the dataset should be extended with features such as Transmission Rate, Transmission Average Time, Transmitted Packets Counts, Total Transmission Time and DIS packets count.
- When BH and SF attacks are triggered, the number of DIO and DAO increases while the number of data packets (i.e., transmitted and received packets) decreases. As a consequence, the same extra features from DR and SH attacks can be used to detect BH attack.

From the four points above, the extra features in Table 5.3 have been calculated per one-second window duration, and have been added to the datasets.

First, we calculate transmitted and received packets counts (SrcCount and DestCount) for each node for 1s duration. Then, we calculate Transmission Rate (TR) and Reception Rate (RR) for each node by dividing SrcCount and DestCount per 1000ms, respectively. We calculate duration time for each packet transmission and reception. Transmission Total Time (TTT) and Reception Total Time (RTT) are calculated by adding up duration time of each transmission and reception packet in 1000ms. Afterwards, Transmission Average Time (ATT) and Reception Average Time (ART) for each node are calculated. Besides, number of transmitted control packets (DAO packets count (DAO), DIS packets count (DIS), and DIO packets counts (DIO)) of each node are calculated within the window size of 1000 ms.

Table 5.3: Features to be used for RPL's intrusions detection

Name	Description
No.	Packet Sequence Number
Time	Simulation Time
Source	Source Node IP
Destination	Destination Node IP
Length	Packet Length
Info	Packet Information
TR	Transmission Rate
RR	Reception Rate
TR/RR	Transmission Rate / Reception Rate
SrcCount	Transmitted Packets Count
DestCount	Received Packets Count
TTT	Transmission Total Time
RTT	Reception Total Time
ATT	Transmission Average Time
ART	Reception Avg. Time
DAO	DAO Packets Count
DIS	DIS Packets Count
DIO	DIO Packets Count
Label	Label takes values 0 or 1

The normal traffic was labelled as 0 while the traffic with malicious behaviour (i.e., each attack related dataset) was labelled as 1. The datasets generated from networks where an attack was triggered were labelled 1 as the entire networks were affected by the malicious activities.

Feature normalisation is used to make convergence quicker and limit the influence of small or large values in the training set, thus increasing the performance of the learning algorithm. We implemented a Python script to mix the normal and malicious datasets for each topology. We firstly applied quantile transformation to the datasets to adjust feature values distribution to normal distribution. We secondly

used min-max scaling to scale all feature values to the range [0,1]. Afterwards, we concatenated all datasets resulting from different topologies (i.e., the three topologies) of each routing attack. As a result, we got six datasets, as detailed in Table 5.4.

Table 5.4: The generated datasets for the IDS use

<b>Datasets</b>	<b>Scenarios</b>	<b>Nb Nodes</b>	<b>Attackers</b>	<b>Packets Counts</b>
Decreased Rank (DR)	DR_25	25	2	503232
	DR_50	50	4	5134640
	DR_100	100	10	7466588
				<b>Total = 13104460</b>
Sinkhole (SH)	SH_25	25	2	513653
	SH_50	50	4	873932
	SH_100	100	10	2735976
				<b>Total = 4123561</b>
Blackhole (BH)	BH_25	25	2	499951
	BH_50	50	4	899333
	BH_100	100	10	6727132
				<b>Total = 8126416</b>
Selective Forwarding (SF)	SF_25	25	2	506444
	SF_50	50	4	891441
	SF_100	100	10	3409921
				<b>Total = 4807806</b>
HelloFlooding (HF)	HF_25	25	2	842548
	HF_50	50	4	2088476
	HF_100	100	10	10263539
				<b>Total = 13194563</b>
Version Number (VN)	VN_25	25	2	2718314
	VN_50	50	4	3585999
	VN_100	100	10	15205283
				<b>Total = 21509596</b>
Multi-Class	MC	25/50/100	2/4/10	<b>Total = 51326396</b>

**5.2.4.2.2 Features Selection.** After the features extraction stage, CVS files are processed to select relevant attributes, which is one of the core concepts in ML. This stage identifies and removes unneeded, irrelevant, weakly relevant, and redundant features from the dataset that do not contribute to the accuracy of the classifier or may decrease its accuracy. In other words, this step would permit to increase accuracy while reducing training time<sup>2</sup> and avoiding bias and model overfitting.

<sup>2</sup>Throughout this chapter, we use the terms "training time" and "fitting time" interchangeably.

There exist several feature selection methods in the literature; the filter methods, the wrapper methods, and the embedded methods. We avoid using wrapper methods because they are computationally costly and are not the most efficient in massive datasets. In this study, we combined an embedded method using the Random Forests classifier, and a filter method applying Pearson correlation, where correlation states how the features are related to each other and to the output variable. Pearson coefficient correlation has a value within 1 and -1, where 1 means total positive linear correlation, -1 means total negative linear correlation and 0 means non-linear correlation. Figure 5.6 summarises the features selection process.

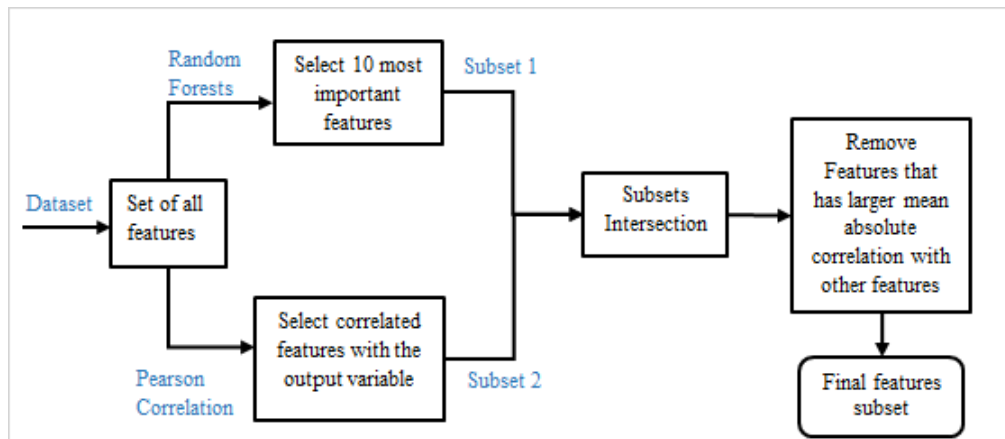


Figure 5.6: Feature selection process.

**Step 1.** We first carried out RF-feature-importance function to highlight the ten most important features for each dataset. When RF classifier is trained, it evaluates each attribute to create splits and gives a score for each feature of the dataset; the higher the score more relevant is the feature towards the output variable (i.e., Label 0 or 1). In this approach, for the tree building process, only a subset of the data samples is chosen with replacement, which is known as bootstrap aggregating or bagging. Nevertheless, this is a biased approach, as it tends to inflate the importance of continuous features or high-cardinality categorical variables. To reduce selection bias, we used cross-validation for feature selection, as reported in [154]. In the cross-validation process, the data is splitting into  $k$  equal folds ( $k=5$ ). The model is trained on  $k-1$  folds and evaluated on the remaining holdout fold. These two steps are performed  $k$  times, each time holding out a different fold. Finally, the performance are aggregated across all  $k$  folds.

In Figure 5.7, the red bars depict the features' importance of the Forests, along-with their inter-trees variability for the Selective-Forwarding dataset, where the x-axis represents the features indexes, and the y-axis represents the importance values.

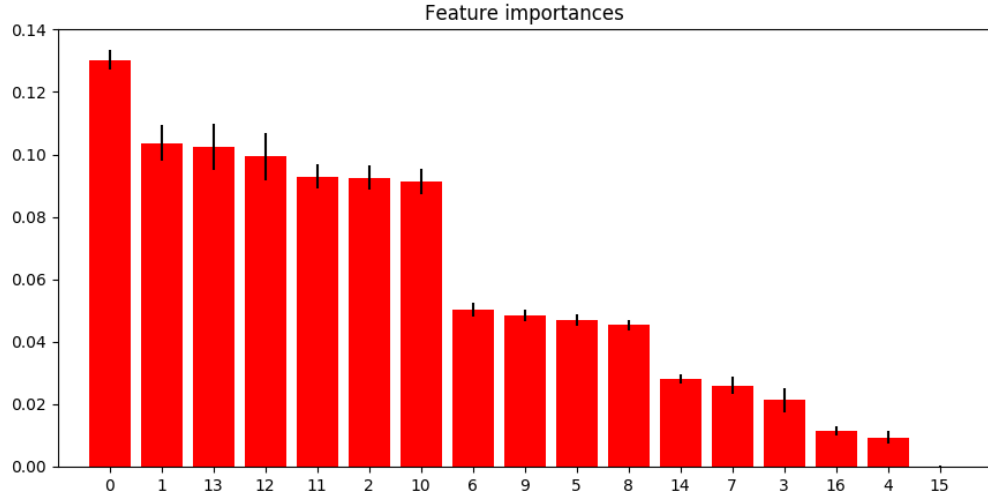


Figure 5.7: Feature importance datagram for SF dataset.

**Step 2.** Secondly, we applied the correlation matrix using the Pearson correlation method on the original features set. Figure 5.8 portrays the correlation matrix for the Selective-Forwarding dataset. We checked the correlation of each feature with the output variable, and we selected a subset of features using a threshold of correlation specific for every attack dataset.

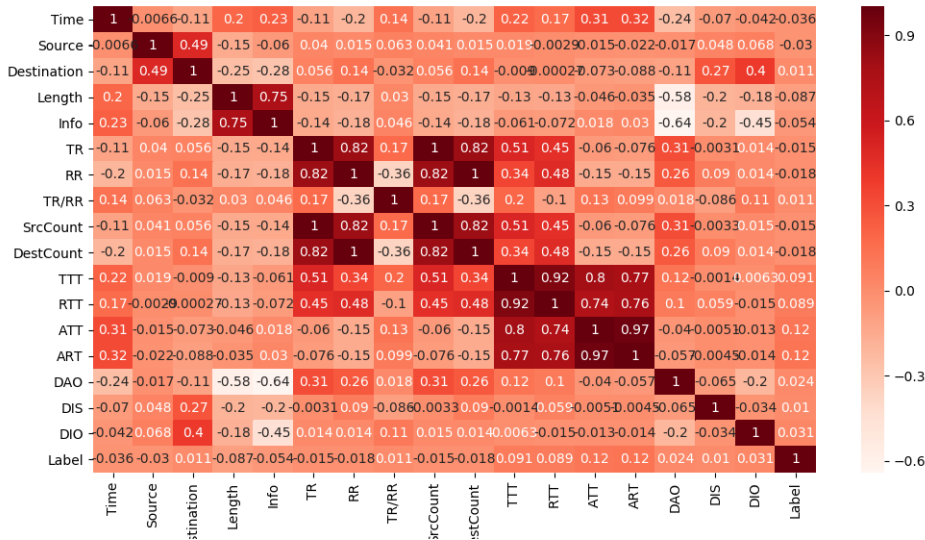


Figure 5.8: Correlations between different features for the SF dataset.

**Step 3.** Afterwards, we selected a new subset of features that represents the intersection of both subsets from the previous two steps.

**Step 4.** According to [155], redundant features should be eliminated since they affect the speed and the accuracy of learning algorithms. Consequently, in the

final step, we checked the correlation of selected features subset with each other using a threshold of 0.8 for the correlation. If these attributes are correlated with each other, we kept only one of them and dropped the rest.

Initially, in each dataset, there are 17 features: Time, Source, Destination, Length, Info, TR, RR, TR/RR, SrcCount, DestCount, TTT, RTT, ATT, ART, DAO, DIS, DIO. After performing the steps mentioned above, the total number of attributes is reduced for each dataset, as presented in Table 5.5.

Table 5.5: Features per dataset after data pre-processing

Datasets	Features	Count	Discard
DR	DAO, Length, TTT, RR, Dst, TR, Src	7	10
SH	Dst, Time, TAR, ATT, Src, RTT, TTT, TR/RR, RR, TR	10	7
BH	Dst, Length, RR, DIS, DIO, TR/RR, TTT	7	10
SF	Time, ART, ATT, Src, RTT, TTT, Dst, RR, TR, DAO	10	7
HF	RR, DIS, DAO, Length, Dst, TR	6	11
VN	ART, RR, TR/RR, Dst, RTT, TTT, DAO	7	10
Multi-class	Dst, Time, ART, ATT, Src, RTT, TTT, TR/RR, RR, TR, DAO, DIO, DIS	13	4

### 5.2.4.3 Multi-class Dataset Generation

To generate a multi-class dataset, firstly, we performed the features extraction steps from Section 5.2.4.2. Secondly, we labelled the normal traffic as 0, and the traffic with malicious behaviour as 1, 2, 3, 4, 5, and 6 for BH, SH, HF, DR, VN, and SF, respectively. Afterwards, we performed the features transformation steps. We mixed the normal and malicious datasets of all attacks for each topology. We then applied quantile transformation to the datasets to adjust feature values distribution to normal distribution. We used min-max scaling to scale all feature values to the range [0, 1]. Next, we concatenated all datasets resulting from different topologies and got one 7-class dataset for all attacks and topologies, as in Table 5.4. Finally, we performed the features selection steps and got the ones in Table 5.5.

### 5.2.5 Classifiers Evaluation and Discussion

To determine the best performing algorithm to classify RPL routing attacks using our datasets, we evaluated the performance of the ML and DL algorithms for 2-class (i.e., normal and attack) and 7-class (i.e., normal and six attacks) datasets.

5.2.5.1 Two-Class Classification Results

For ML models, we applied 5-fold cross-validation on the datasets. For the DL model, we used 90% of the data for training and 10% for model evaluation. We saved all trained models for possible future use. We assessed the accuracy, precision, recall, and f1\_score performance metrics for each algorithm and obtained the results on each dataset, as presented in Figure 5.9.

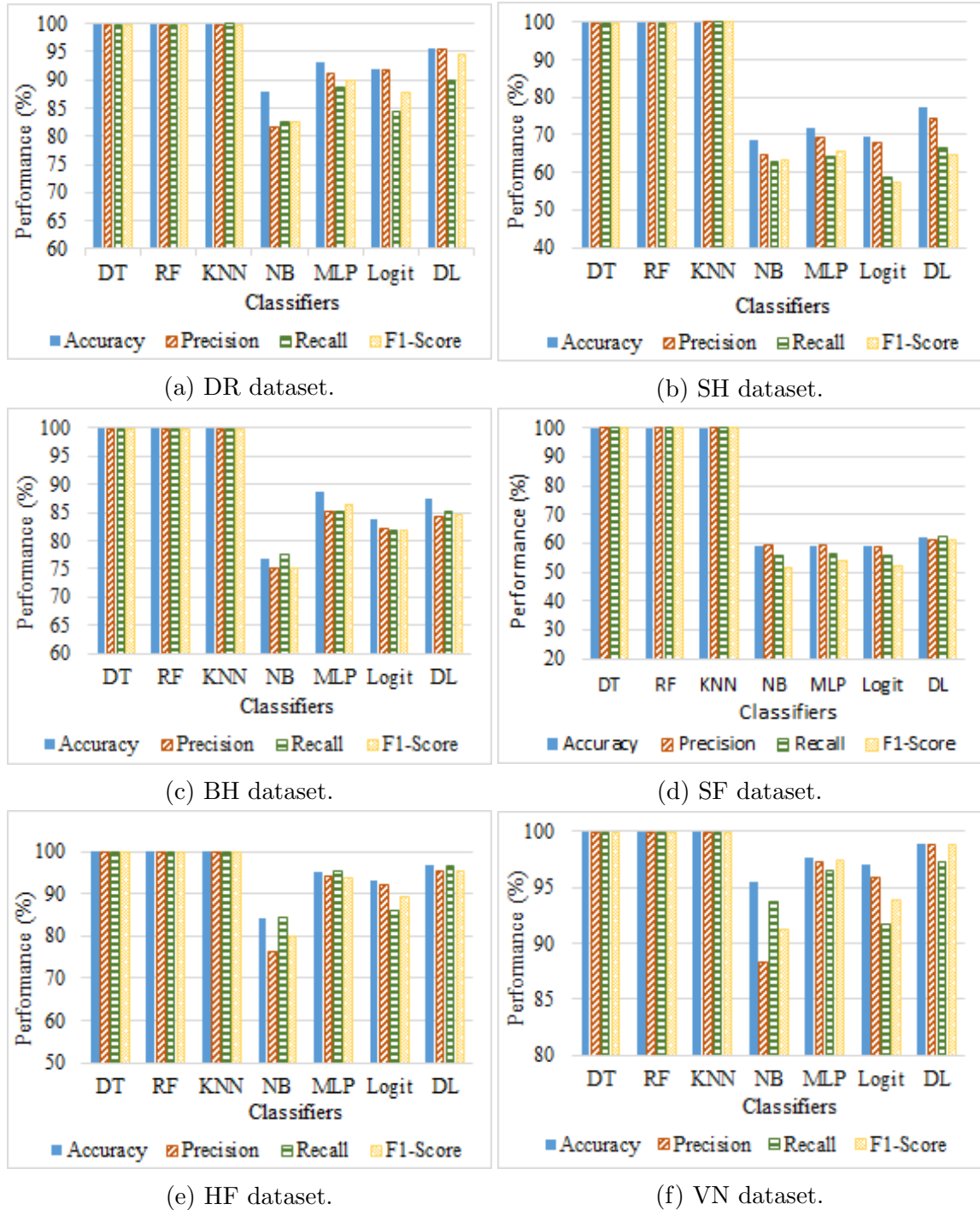


Figure 5.9: Classifiers performance per dataset for 60 minutes simulation time.



We observe that DT, RF and KNN showed better performance for all metrics as compared to NB, MLP, Logit, and DL. We also notice that the same three algorithms achieved a classification accuracy rate of more than 99% for both 2-class and 7-class classifications, as shown in Figure 5.9 and Figure 5.13.

Nevertheless, compared with DT and RF, KNN is very slow to converge and gets significantly slower as the number of independent variables increases (i.e., the dataset size increases), as shown in Figure 5.10. Besides, the MLP classifier also required a longer fitting time but gave better performance when the size of the dataset decreases, as can be seen in Figure 5.11. On the other hand, the DL classifier takes the longest training time, and the training time increases with the dataset size.

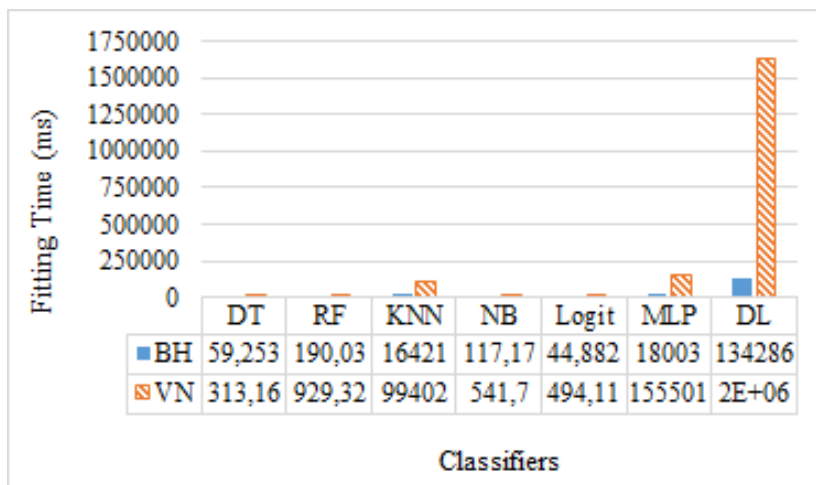


Figure 5.10: Fitting time for BH dataset vs VN dataset for 10mn simulation time.

When we compare the two ANN-based classifiers (i.e., the MLP classifier with one hidden layer and 100 neurons, and the DL model in Figure 5.9), we find that for almost all datasets, the latter gives slightly better performance except for BH attack where the former gives relatively better results (88.56% vs 87.6%). These results are due to the fact that the DL model has more capacity than MLP (i.e., the number of layers and neurons in the DL model is higher than in MLP classifier).

From another side, a DL model with increased capacity tends to yield better accuracy up to a point at which the model stops improving [156]. As an example, Figure 5.12a and Figure 5.12b draw the DL model's loss and accuracy, respectively, of the VN dataset for 10 minutes' simulation time. We notice that the accuracy of the DL model for a larger dataset (60 minutes' simulation time in Figure 5.9f) is higher compared to a smaller dataset (10 minutes' simulation time in Figure 5.12b). Nevertheless, although DL methods are getting lots of attention lately because of their promising results in several areas, such as signal processing, natural language processing, and image recognition, the biggest the DL model, the more computa-

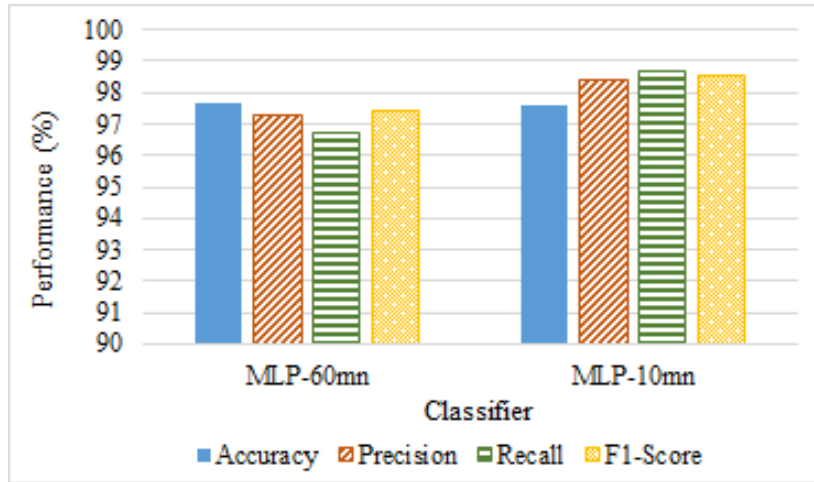


Figure 5.11: MLP performance for VN datasets: 10mn vs 60mn simulation time.

tional resources it requires and the longer it takes to train which is not suitable for intrusion detection in IoT networks. Furthermore, the present evaluation results confirm that the DL methods are not desirable for intrusion detection for IoT networks.

### 5.2.5.2 Multi-Class Classification Results

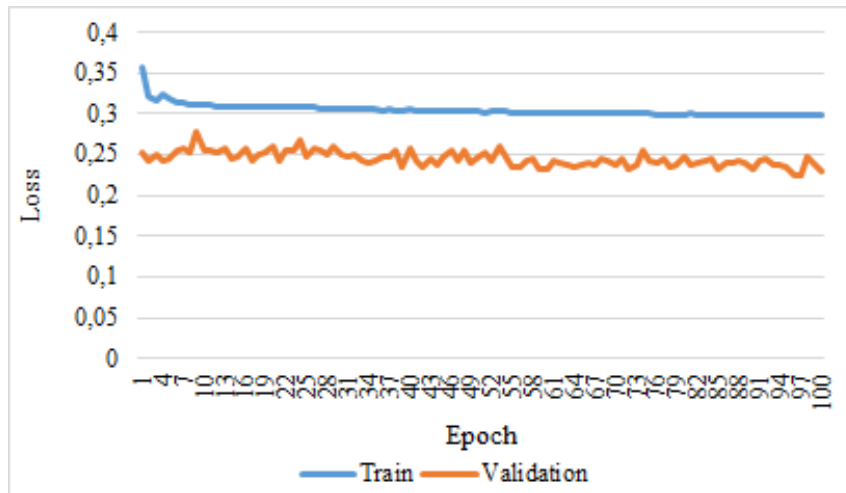
We used the multi-class dataset in Table 5.4. Because the 7-class dataset is too large (51326396 packets), we shuffled it and used half of the generated dataset to test the ML and DL classifiers. We got the performance plotted in Figure 5.13. The results show that KNN has an accuracy of 99% with a detection rate of 98%. RF and DT take the second position with an accuracy of 98% and a detection rate of 98%. On one other hand, compared with 2-class classification, MPL, Logit, and NB gave a mediocre performance with precision, recall and F1-score around 35%. Regarding DL, the model did not converge after three weeks of execution, which makes it impractical as IDS for IoT networks, especially for real-time needs.

From the obtained results, we conclude that in terms of performance and fitting time, RF is more suitable for intrusion detection for RPL-based networks.

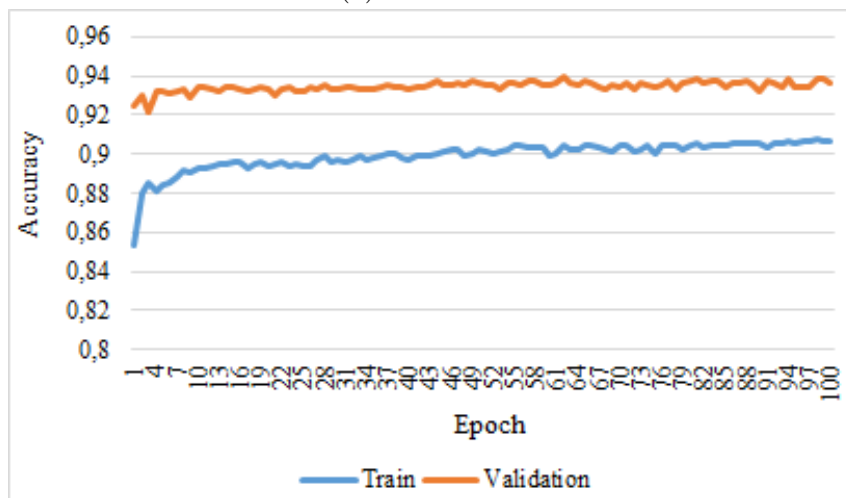
## 5.2.6 RF-Based Intrusion Detection System for RPL (RF-IDS<sub>R</sub>)

### 5.2.6.1 System Model and Assumption

As presented in Chapter 2, IoT networks play an important role in the establishment of Industry 4.0 and other daily human's applications and thus need to be secure. Because RPL is the de facto routing protocol for IoT networks, we introduce an AI-driven IDS for RPL, namely, RF-IDS<sub>R</sub>. RF-IDS<sub>R</sub> is an anomaly-based IDS that uses



(a) Model loss.



(b) Model accuracy.

Figure 5.12: DL-model for VN dataset.

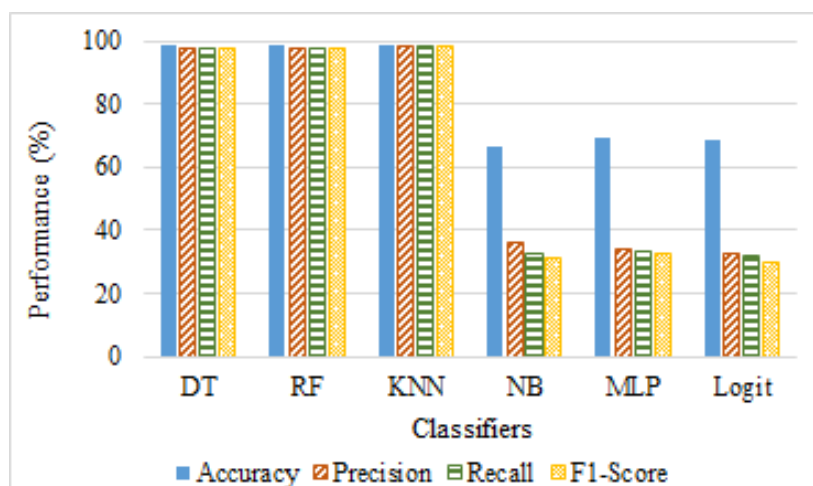


Figure 5.13: 7-class classification performance.

the Random Forest model (see Section 2.5) to detect routing attacks. Figure 5.14 depicts a graphical representation of the IDS architecture. RF-IDSR is a hybrid-IDS

(i.e., distributed and centralised) relying on the collaboration of three actors: the DODAG border router (BR) or edge node, the Monitoring Nodes (MNs), and the sensor nodes. The following assumptions are considered.

- The BR is a powerful node, which has not processing and energy consumption constraints.
- The BR is always trusted and cannot be compromised by an adversary.
- The BR shares a secret key with each node to encrypt and secure data packets.
- Two RPL instances coexist:
  - The first instance, called sensor network (SN), is an RPL-based network composed of both resource-constrained sensor devices and more powerful devices. The SN is used to route the sensed data to the BR. Independently of the RF-IDSR, to prevent attacks and failures, each sensor node implements lightweight appending for the RPL protocol, as presented in Section 5.2.6.3.
  - The second instance, called the monitoring network, is composed of a few more powerful nodes (i.e., monitoring nodes). The MNs are powerful machines and devices, which have not resource-constraints. Notably, the MNs do not have battery depletion issues.
- We assume the MNs are synchronised with each other and with the BR. The MNs are selected based on the geographical location of the nodes in the first RPL instance. Indeed, the architecture can be seen as a set of virtual clusters of nodes from both RPL instances, where MNs are cluster heads with enough resources (e.g., energy power) for intrusion detection purpose.

### 5.2.6.2 Attacks Detection using RF Model

One objective of this work is to create a predictive model to classify the RPL-based network packets into two classes: Normal or Attack and identify the attack using the multi-class dataset. In this paper, we select RF as the classifier to be used to detect RPL routing attacks because of its high accuracy of prediction, computational and time efficiency, and its ability to select features according to their importance [64]. Indeed, in the next sections we demonstrate that the RF model gives better results compared to other classifiers. Besides, compared to KNN (see Section 2.5) that gives better accuracy than RF, the latter returns predictions in a shorter time, especially for large datasets (see Section 5.2.5). As presented in Figure 5.14, RF-IDSR is composed of three modules defined as follows.

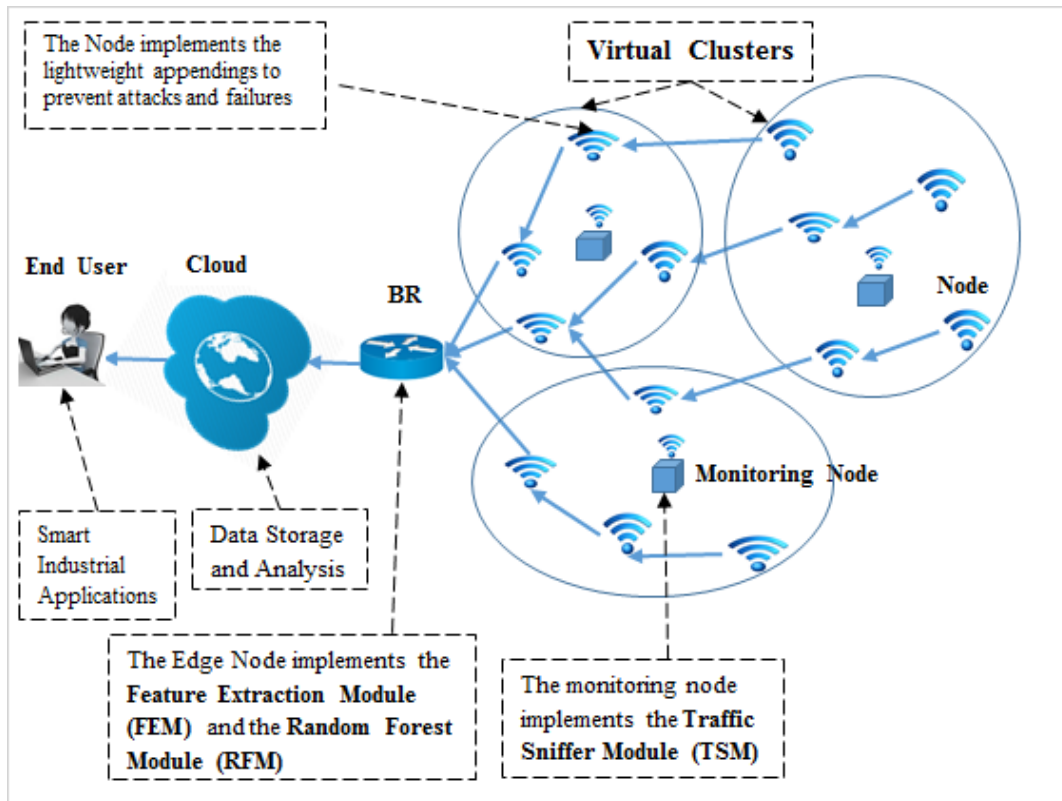


Figure 5.14: RF-IDS architecture.

- Distributed Module: is a module placed at each MN of the second RPL instance.
  1. Traffic Sniffer Module (TSM): TSM implements Algorithm 3. The MNs use TSM to sniff the traffic from the first RPL instance. MNs must timestamp packets as they are received by their radio, generate PCAP files each 1-minute window (one PCAP file per MN), and send the PCAP files to the BR through the second RPL instance paths.

---

**Algorithm 3** Monitoring Algorithm

---

1. Sniff packets
  2. Timestamp packets
  3. Generate a PCAP file for the last 1-minute window of the sniffed packets
  4. Send the PCAP file to the BR
- return** PCAP
- 

- Centralised Modules: are modules placed at the BR.
  1. Feature Extraction Module (FEM): FEM implements Algorithm 4. FEM allows the BR to gather all received PCAP files from the MNs, concatenate them, and process them to extract features and generate new data

(equivalent to the testing data in the dataset) as presented in Section 5.2.4.2. The BR may use the cloud to store and process the data.

---

**Algorithm 4** Datasets Generation Algorithm

---

**Require:** PCAP files received from MNs

1. New-PCAP = Concatenate PCAP files
2. New-PCAP = Delete duplicate packets from New-PCAP
3. dataset.csv = Ttransform New-PCAP to CSV format
4. New-Data = Execute steps in Section 5.2.4.2
  - a. Sorting dataset.csv by Time attribute
  - b. Feature Transformation (Source, Destination, Info, etc.)
  - c. Feature Extraction within 1-second window size
  - d. Feature Selection as summarised in Figure 5.6
5. Store New-Data in the cloud (optional)

**return** New-Data

---

2. Random Forests Module (RFM): RFM implements the trained RF model (from Section 4.6) and Algorithm 5. The BR uses RFM to evaluate the new data generated from FEM using the trained RF model and gives predictions. Furthermore, RFM implements an update process to update, periodically, the RF model using the new data from FEM. The last point permits to enrich the learning algorithm, and thus to detect new threats. RFM may use the cloud to update the RF model and execute Algorithm 5 (see Figure 5.14). The alarm may be sent to the end-user (e.g., the network administrator) and notifications to the sensor nodes to discard the malicious nodes from participating in the network operations.

---

**Algorithm 5** Anomaly Detection Algorithm

---

**Require:** New-Data from FEM

1. Load trained RF model
2. Scores = Predict outcomes on New-Data
5. If Intrusion, raise an alarm and send notifications

**return** Scores

---

### 5.2.6.3 Attacks and Failure Prevention

In addition to the presented RF-IDSR, we propose considering three lightweight appending to RPL aiming to prevent and tolerate the HelloFlooding, version number, and global repair attacks, as well as network failure.

**5.2.6.3.1 HelloFlooding, DIS , and SybM Attacks Prevention.** HelloFlooding, DIS and SybM attacks can be prevented and tolerated relying on our contribution in Chapter 4-Section 4.5.

### 5.2.6.3.2 Global Repair and Version Number Attacks Prevention and

**Detection.** The BR increments the version number in case of inconsistency. This leads to the initiation of a global repair and the DODAG is rebuilt from scratch. A malicious node can post a false version number in its control message to force a global repair. Unfortunately, there is no security mechanism currently available in RPL to protect it against this intrusion or to check the integrity of the version number field.

There exist some work that propose solutions to detect the VN attack; nevertheless, they rely on statistics and other nodes verifications to detect or prevent the intrusion. Authors in [87] propose a double authentication scheme for the version number and the Rank field; however, they did not gave a simulation based evaluation of their proposal.

To tolerate and detect the version number and global repair attacks, we authenticate the DODAG version field of the DIO message. In our approach, the BR uses a one-way hash chain to generate sequence numbers for the DODAG version field. A one-way hash chain is a sequence of numbers,  $V_i$  ( $0 \leq i \leq n$ ), generated by a one-way hash function  $F$  as in Equation 5.6, where  $V_i$  is a random number generated by the BR, and the  $F$  function is the same for the BR and all nodes in the network.

$$\forall i, 0 \leq i < n : V_i = F(V_{i+1}) \quad (5.6)$$

In our approach, the one-way hash chain is stocked in the BR. In addition, the first value of the one-way hash chain used to generate the DODAG versions is uploaded securely into the nodes before deployment. Besides, when a new node is deployed in the network, it is pre-loaded with the first unused value of the chain.

In this work, every global repair is identified with a DODAG version ( $V_i$ ) that is the last delivered value of the one-way hash chain. Consequently, the nodes of the network can verify the new DODAG version  $V_{i+1}$  through checking whether  $V_i = F(V_{i+1})$ , where  $V_i$  is the previous DODAG version. On the other hand, the nodes cannot calculate the following DODAG version since the security of the one-way hash chain concept is based on the fact that knowing  $V_i$ , it is computationally infeasible to determine  $V_{i+1}$ .

In the DIO, the version number is an 8-bit unsigned integer. As our solution is hash-based, we know that the larger the size of the hash result, the more difficult is for the attacker to break the hash. Therefore, we extended the size of the version number to 32 bits. Furthermore, we chose to use multiple hashing algorithms simultaneously and randomised the number of rotation of the calculations to make the solution difficult to break.

The pseudocode in Algorithm 6 summarises the proposed solution.

---

**Algorithm 6** Global Repair and Version Number Attacks Prevention

---

**Require:**  $V_0$  (i.e., the first value of the hash chain uploaded in all nodes before deployment)  $V_i$  (i.e., the DODAG version of the  $i$ th global repair and the last delivered value of the one-way hash chain);  $F$  (i.e., the one-way hash function implemented in both the BR and the in-network nodes);  $V_i$  ( $0 \leq i \leq n$ ) (i.e., the one-way hash chain stocked in the BR);

**if** a node receives a DIO with a new DODAG version value  $V_{i+1}$  knowing that  $V_i$  is the current DODAG version **then**

The node calculates  $F(V_{i+1})$

**if**  $V_i = F(V_{i+1})$  **then**

(the node is sure that the BR updated the DODAG version)

The node reinitialises its Trickle timer

It updates the DODAG version field to  $V_{i+1}$  and broadcasts a DIO message with the new version number

**else**

The node discards the received DIO and considers the node from which it receives the DIO as malicious

**end if**

**end if**

---

We simulated a network of 30 zolertia nodes (Z1 motes) with one BR and 29 senders placed randomly. Actually, the Z1 resources, although limited, are sufficient for our solution. Table 5.6 shows the simulation parameters.

Table 5.6: Simulation parameters for VN attack tolerance and detection

Parameter	Value
Simulator	Cooja-Contiki 3.0
Simulation time (mn)	10
Number of nodes	30
Network area	$100 \times 100 m^2$
Transmission range	35m
Interference range	55m
TX, RX	100 %, 80 %
Radio medium	UDGM : Distance Loss
Traffic rate	1 packet sent every 60 seconds
Attacker nodes	2

We simulated six scenarios; (1) native RPL, (2) RPL with solution (RPL+sol), (3) RPL with global repair (RPL+GR), (4) RPL with solution and global repair (RPL+Sol+GR), (5) RPL under VN attack (RPL+VNA), and (6) RPL with solution under VN attack (RPL+Sol+VNA). Like in RPL-MRC evaluation, we used PDR, control overhead and the average energy consumption as performance metrics to evaluate our approach.

- **Packet Delivery Ratio.** The results obtained in Figure 5.15 show that the



PDR takes almost identical values in the first four scenarios, which means that our solution has no negative effect on RPL's PDR in the absence of the intrusion. However, when the VN attack is in place, we can see that the PDR starts to drop to 32% in the scenario without solution. This is evident as the attackers cause packets to be dropped or lost due to various reasons such as packet collision, link break, etc. On the other hand, with the solution, almost all the packages were delivered to the BR successfully.

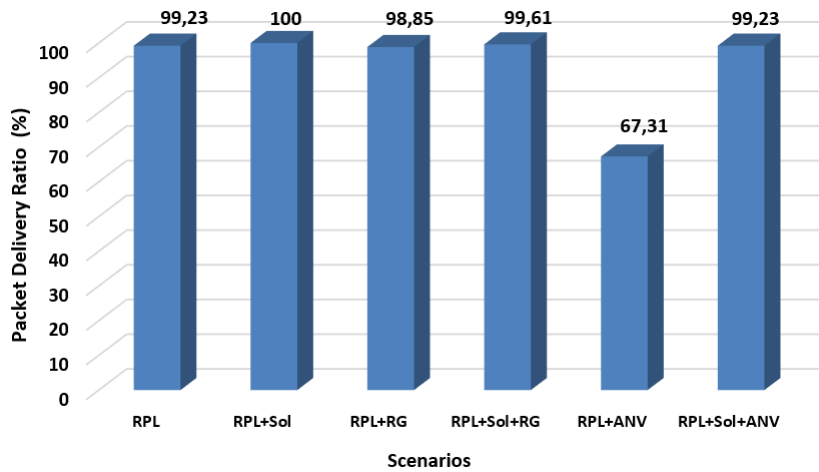


Figure 5.15: Packet delivery ratio vs scenarios.

- Control Overhead.** Referring to Figure 5.16, the control messages overhead in the third and fourth scenarios increases when a global repair is initiated, which is normal as the nodes in the network broadcast DIOs to reconstruct the DODAG graph. In the presence of malicious nodes, the overhead increases by 2346 messages due to the characteristics of the version number attack. In contrast, the proposed scheme prevents the diffusion of suspect DIOs to reconstruct DODAG, therefore, the control overhead is much less than that of RPL under VN attack.

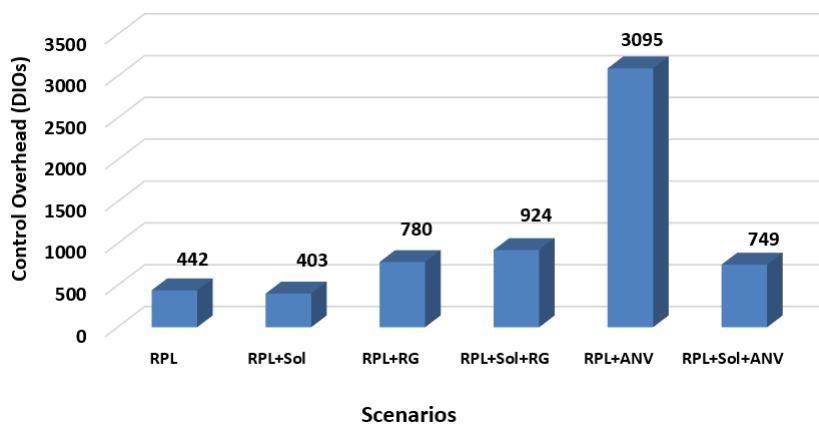


Figure 5.16: Control overhead vs scenarios.

We did another more in-depth study of this metric for the neighbouring nodes of the malicious node (in this case the node 17). The number of outgoing DIO messages from each neighbour for the three scenarios RPL, RPL+VNA and RPL+Sol+VNA, is shown in Figure 5.17. We can observe that as soon as the node 17 turns into an attacker, the overhead produced by its neighbours can increase up to 3 times within a short time (i.e., 3 mn). This causes a significant increase in control packets across the network. Indeed, we can clearly see that our approach was able to reduce the number of DIO messages transmitted by the attacker's neighbours. Actually, our approche counters the intruders and discard them from participating in the network operations and thus stabilises the control messages transmission.

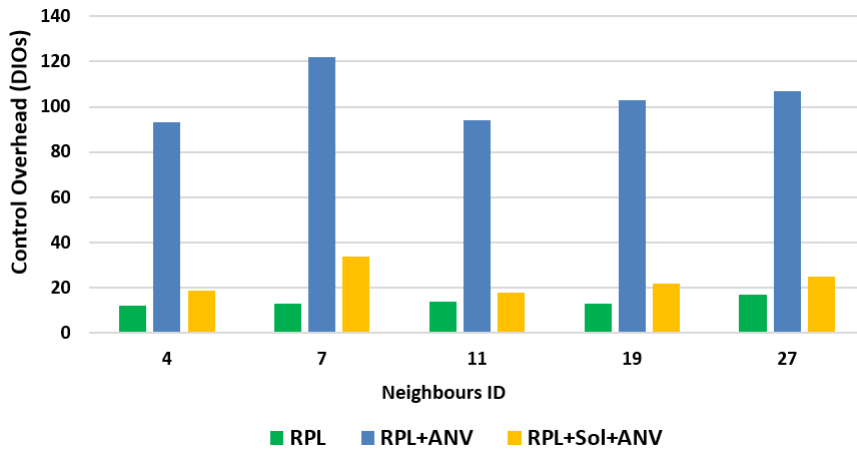


Figure 5.17: Control overhead for the neighbours of the intruder 17.

- Average Power Consumption.** The average energy consumption for the different scenarios is shown in Figure 5.18. From the second and fourth scenarios, we conclude that the hash functions consumed very little power making the obtained results acceptable. In the case of RPL under attack, we observe that VN attack can significantly affect the power consumption of the nodes and reduce their life time. Nevertheless, in the case of RPL with solution and under attack, we can see that the energy consumed is lower, which allows us to say that our approach has been successful in tolerating and detecting the intrusion while conserving the energy.

Besides, Figure 5.19 shows that the neighbours of the malicious node 17 consumed little power to discover and stop the attacker. As a result, the VN attack did not affect their power consumption.

**5.2.6.3.3 Fault and Intrusion Tolerance.** In RPL, a preferred parent (PP) is used by children nodes to forward traffic until detection of a better route, a path

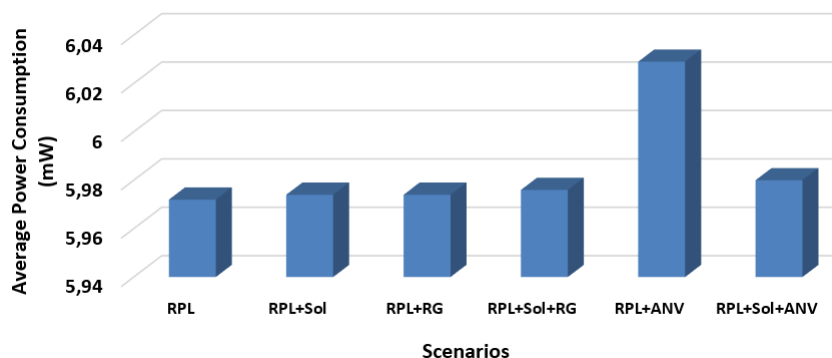


Figure 5.18: Average power consumption vs scenarios.

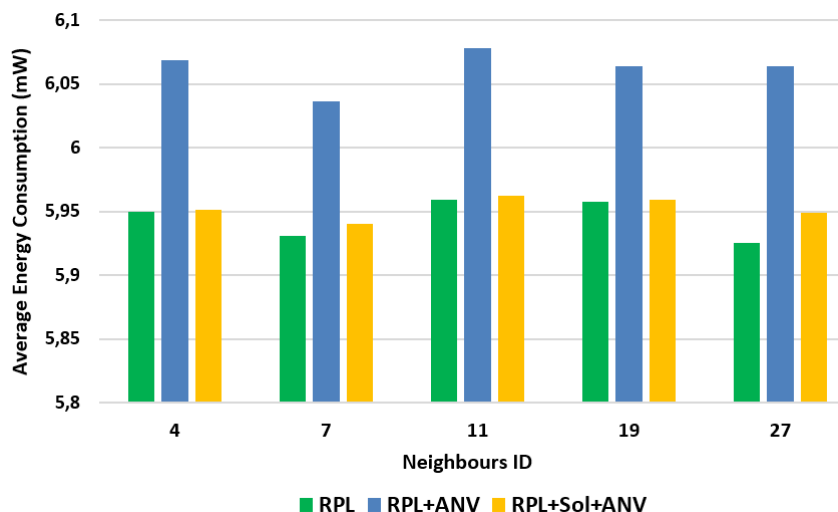


Figure 5.19: Average power consumption for the neighbours of the intruder 17.

failure or an intrusion (i.e., the PP is detected as a malicious node). Thus, other potential parents are rarely used. To enhance the resiliency and security of RPL, our approach merges intrusion-tolerance with fault-tolerance solutions by considering a multi-path strategy for RPL. We propose to redefine the RPL's objective function in a way to select two random parents through which the traffic is routed. Choosing only two paths reduces the network overhead while increasing the packet delivery ratio and preventing attacks, such as Wormhole, blackhole, selective forwarding and sinkhole. Indeed, a packet at each hop is routed through two potential parents. On the one hand, the parents are selected randomly, and thus, cannot be specifically targeted by an attacker; because, the attacker has no way to know the parents that will forward the traffic. On the other hand, our solution prevents path and node failure because, at each hop, the traffic is forwarded through two random parents.

In RPL, nodes ignore any DIO message from nodes of higher or equal ranks aiming to avoid loops [22]. However, in some cases, a node could have only one or two potential parents. To expand the parents' list and the selection choice, if the number of potential parents is less than or equal to two, our solution adds nodes

with the same rank in the parents' list as supplementary parents.

### 5.3 Summary

In this chapter, as a first step, we introduced a new Trust-based IDS (T-IDS) to deal with the mobility, identity and security gaps of the RPL routing protocol. T-IDS is hierarchical where three layers cooperate to handle the routing attacks: the backbone station, the border router, and the in-network nodes. Furthermore, T-IDS uses three modules: IdentityMod, MobilityMod, and IDSMoD to detect and avoid malicious nodes. We presented a demonstrative algorithm to show how T-IDS can deal with SybM attack introduced in Chapter 4. Even if T-IDS seems to be resources costly, we believe that off-loading security computations and data-storage using TPM reduces the cost. Nevertheless, T-IDS is not able to detect new attacks.

To overcome the above-cited inconvenient, as a second step, we studied the applicability of ML and DL techniques for intrusion detection in RPL-based IoT networks. We demonstrated that with the selection of the appropriate features, high performance had been achieved. In the 2-class classification, the decision tree (DT), random forests (RF), and K-Nearest Neighbours (KNN) classifiers recorded more than 99% for each of the following metrics: accuracy, precision, recall, and f1-score. The recorded detection rate (Recall), precision, and f1-score for multi-class classification were more than 98% for the three classifiers, while the KNN accuracy was 99%. Besides, RF recorded the lowest fitting time. On the other hand, the DL model, MLP, Naïve Bayes (NB), and Logistic Regression (LR) classifiers recorded lower performance.

The evaluation results showed that RF is a good classifier for RPL networks threats detection. Consequently, we introduced the RF-based IDS, named RF-IDSR, to provide both fault tolerance and intrusion tolerance and detection for RPL LLNs. RF-IDSR uses RF classifier to categorise RPL-based attacks using a multi-class dataset. Furthermore, we presented lightweight appending to RPL to prevent (tolerate) the HelloFlooding, version number, global repair attacks, and network failure. The mechanism for HelloFlooding tolerance has been introduced in Chapter 4.

# Chapter 6

## Conclusions and Perspectives

### 6.1 Thesis Summary and Contributions Review

The first objective of this thesis has been to consider the security fault tolerance aspect of the Internet of Things (IoT) environment as it is profoundly inclined to threats and faults dangers. Because the intrusions are considered as faults, fault tolerance in the context of IoT security consists on making: i) all nodes secure by default, ii) all nodes able to know the state of the network, and iii) protocols such as the routing protocol embody intrusion detection systems (IDSs) and tolerance mechanisms to ward off attacks.

Therefore, the main focus of this thesis has been to introduce new IDSs and intrusion tolerance schemes to enhance the security of the de-facto routing protocol, named IPv6 Routing Protocol over Low Power and Lossy Networks (RPL), which has been standardised for IoT Low-Power and Lossy Network (LLNs). LLNs are considered as a key enabling component of IoT. RPL effectively organises and maintains the IoT-LLNs taking into consideration limitations of such networks. Though RPL provides some cryptographic security features aiming to counter external attacks, it is vulnerable to many insider attacks. It has been shown in the literature that the attacks that exploit RPL's control messages, rules, and operations have evolved in terms of diversity, thus disrupting the established LLNs routes, causing normal devices perform heavy computations, degrading the network performance, and shortening the network lifetime often resulting in denial of service. Subsequently, more attention needed to be paid to the analysis of these intrusions and their detection methods.

The first objective of the thesis was to have a solid background on the IoT, IDSs, and machine learning (ML) concepts, which has been achieved, as documented in Chapter 2. The chapter presented the IoT's definitions and applications, characteristics and standardised protocols, and its security challenges. In addition, it provided

the preliminary information about the definitions relevant to IDSs, the different types of IDSs and their detection methods. Finally, the chapter elaborated on the ML concepts and the algorithms studied in this thesis.

The second objective was to analyse the security issues of the RPL protocol, as RPL is the main focus of this thesis. Therefore, more details and discussions have been given in Chapter 3 to usher the essential background for better understanding the research problems within this thesis. In consequence, a variety of defence solutions for RPL were surveyed, especially IDSs as they are a relevant point for the security fault tolerance.

Based on the literature review from Chapter 3, a security issue related to the gaps in mobility and identity management in RPL was identified. Accordingly, an analytical study of the RPL performance under a new Sybil-mobile attack, named SybM attack, was achieved. Hence, the third objective was to implement SybM attack and evaluate the RPL performance with simulations. The fourth objective was to develop a new mitigation mechanism for RPL to tolerate SybM and multicast related attacks, named RPL-MRC. In RPL-MRC, RPL itself is adapted to reduce the response to multicast messages in static or dynamic networks regardless of the sender node identity. The solution is promising as it has achieved high performance by reducing significantly the control overhead, power consumption, and data packet overhead. These objectives were documented in Chapter 4

The state-of-the-art from Chapters 3 and 4 revealed the significant consequences of vulnerabilities on RPL, and the need for more solutions in this regards. Consequently, the last objective was to propose IDSs solutions for RPL, which is documented in Chapter 5. Hence, our first contribution to secure RPL was to introduce T-IDS a cross-layer trust-based IDS that relies on the RPL specification to detect intrusions. It uses specific modules to counter mobility-based and identity-based attacks like SybM attack. Besides, it reorganises the network according to the trustworthiness of the participating nodes.

Because T-IDS is not able to detect unknown attacks, our second contribution was to introduce RF-IDSR. RF-IDSR is an ML-based IDS extended with three mechanisms for intrusions tolerance. As its name suggests, RF-IDSR is based on the Random Forests algorithm to detect intrusions. To evaluate the IDS, the RPL attacks that threaten the most its functionalities were implemented, then one-class and multi-class datasets were generated. Several ML classifiers and a deep learning model were implemented and compared to select the one with the best detection rate. The proposed HelloFlooding, DIS , and SybM attacks prevention mechanism is able to reduce the effects of the DIS and SybM attacks as presented in Chapter 4. The global repair and version number attacks prevention mechanisms eliminates the effects of the discussed attacks as demonstrated in the simulations results. Further-

more, the fault and intrusion tolerance mechanism is able of preventing the selective forwarding attack.

## 6.2 Limitations and Future directions

In this section, we outline some limitations of our contributions and propose future directions to address them.

- Our RF-IDSR is only proposed for the six specification-based attacks. It would be interesting to test the model with other RPL related attacks as almost all the attacks impact the same performance metrics (i.e., features) used in the developed datasets.
- As stated above, the created RPL threats datasets aim to detect only six specification-based attacks. It would be interesting to incorporate other threats to the multi-class dataset to allow better training and testing of the detection model. Indeed, with the continuous modification on the attacks' strategies, both the dataset and IDS need to be extended dynamically, which is a challenge that must be investigated. For instance, the dataset could be extended with attacks from other layers, such as the the 6LoWPAN adaptation layer.
- The assembly of RF-IDSR and T-IDS into one IDS will be a promising enhancement for RF-IDSR. Having an RF-based IDS for detecting attacks that is augmented with mobility and identity management modules, a trust-based module for self-organising, and intrusions tolerance mechanisms for reducing or eliminating the attacks' effects on the network, while taking into consideration the limitations of RPL and LLNs is very interesting to investigate.
- Another interesting direction is the deployment and assessment of the proposed intrusion detection and tolerance mechanisms in a real testbed.

# Bibliography

- [1] T. Mohamed, T. Otsuka, T. Ito, Towards machine learning based iot intrusion detection service, in: IEA/AIE, 2018.
- [2] Z. Shelby, C. Bormann, 6LoWPAN: The wireless embedded Internet, Vol. 43, John Wiley & Sons, 2011.
- [3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, *IEEE communications surveys & tutorials* 17 (4) (2015) 2347–2376.
- [4] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, P. Faruki, Network intrusion detection for iot security based on learning techniques, *IEEE Communications Surveys & Tutorials* 21 (3) (2019) 2671–2701.
- [5] O. Iova, P. Picco, T. Istomin, C. Kiraly, Rpl: The routing standard for the internet of things... or is it?, *IEEE Communications Magazine* 54 (12) (2016) 16–22.
- [6] M. B. Yassein, S. Aljawarneh, E. Masa'deh, A new elastic trickle timer algorithm for internet of things, *Journal of Network and Computer Applications* 89 (2017) 38 – 47, emerging Services for Internet of Things (IoT). doi:<https://doi.org/10.1016/j.jnca.2017.01.024>.
- [7] A. Fraboulet, G. Chelius, E. Fleury, Worldsens: development and prototyping tools for application specific wireless sensors networks, in: Proceedings of the 6th international conference on Information processing in sensor networks, 2007, pp. 176–185.
- [8] B. Djamaa, Pervasive service discovery in low-power and lossy networks (2016).
- [9] A. Dunkels, J. Eriksson, N. Finne, N. Tsiftes, Powertrace: Network-level power profiling for low-power wireless networks (2011).
- [10] X. Jia, Q. Feng, T. Fan, Q. Lei, Rfid technology and its applications in internet of things (iot), in: 2012 2nd international conference on consumer electronics, communications and networks (CECNet), IEEE, 2012, pp. 1282–1285.



- [11] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, Internet of things: Vision, applications and research challenges, *Ad hoc networks* 10 (7) (2012) 1497–1516.
- [12] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (iot): A vision, architectural elements, and future directions, *Future Generation Computer Systems* 29 (7) (2013) 1645–1660.
- [13] N. Kushalnagar, G. Montenegro, C. Schumacher, et al., Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals (2007).
- [14] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, et al., Transmission of ipv6 packets over ieee 802.15. 4 networks, *Internet proposed standard RFC 4944* (2007) 130.
- [15] Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans), *IEEE Std 802.15.4-2011* (Revision of IEEE Std 802.15.4-2006) (2011) 1–314doi:10.1109/IEEESTD.2011.6012487.
- [16] J. Hui, P. Thubert, et al., Compression format for ipv6 datagrams over ieee 802.15. 4-based networks (2011).
- [17] J. Hui, J. Vasseur, D. Culler, V. Manral, An ipv6 routing header for source routes with the routing protocol for low-power and lossy networks (rpl), *Request for Comments 6554* (2012).
- [18] Routing over low power and lossy networks (roll).  
URL <https://datatracker.ietf.org/wg/roll/charter/>
- [19] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, Collection tree protocol, in: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys '09*, ACM, 2009, p. 1–14. doi:10.1145/1644038.1644040.
- [20] S. Dawson-Haggerty, A. Tavakoli, D. Culler, Hydro: A hybrid routing protocol for low-power and lossy networks, in: *2010 First IEEE International Conference on Smart Grid Communications*, 2010, pp. 268–273. doi:10.1109/SMARTGRID.2010.5622053.
- [21] J. Vasseur, N. Agarwal, J. Hui, Z. Shelby, P. Bertrand, C. Chauvenet, Rpl: The ip routing protocol designed for low power and lossy networks, *Internet Protocol for Smart Objects (IPSO) Alliance* 36 (2011).

- [22] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, Rpl: Ipv6 routing protocol for low-power and lossy networks, RFC 6550, Internet Engineering Task Force (2012).
- [23] R. Roman, J. Zhou, J. Lopez, On the features and challenges of security and privacy in distributed internet of things, *Computer Networks* 57 (10) (2013) 2266–2279.
- [24] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, M. Richardson, A security threat analysis for the routing protocol for low-power and lossy networks (rpls), Tech. rep. (2015).
- [25] P. Pongle, G. Chavan, A survey: Attacks on rpl and 6lowpan in iot, in: 2015 International conference on pervasive computing (ICPC), IEEE, 2015, pp. 1–6.
- [26] A. Mayzaud, R. Badonnel, I. Chrisment, A taxonomy of attacks in rpl-based internet of things, *International Journal of Network Security* (2016).
- [27] F. Medjek, D. Tandjaoui, I. Romdhani, N. Djedjig, Security threats in the internet of things: Rpl’s attacks and countermeasures, in: *Security and privacy in smart sensor networks*, IGI Global, 2018, pp. 147–178.
- [28] F. Medjek, D. Tandjaoui, M. R. Abdmeziem, N. Djedjig, Analytical evaluation of the impacts of sybil attacks against rpl under mobility, in: 2015 12th International Symposium on Programming and Systems (ISPS), IEEE, 2015, pp. 1–9.
- [29] F. Medjek, D. Tandjaoui, I. Romdhani, N. Djedjig, Performance evaluation of rpl protocol under mobile sybil attacks, in: 2017 IEEE Trustcom/BigDataSE/ICISS, 2017, pp. 1049–1055. doi:10.1109/Trustcom/BigDataSE/ICISS.2017.351.
- [30] F. Medjek, D. Tandjaoui, I. Romdhani, N. Djedjig, A trust-based intrusion detection system for mobile rpl based networks, in: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2017, pp. 735–742.
- [31] A. Andrushevich, B. Copigneaux, R. Kistler, A. Kurbatski, F. Le Gall, A. Klapproth, Leveraging multi-domain links via the internet of things, in: *Internet of Things, Smart Spaces, and Next Generation Networking*, Springer, 2013, pp. 13–24.

- [32] R. Khan, S. U. Khan, R. Zaheer, S. Khan, Future internet: the internet of things architecture, possible applications and key challenges, in: 2012 10th international conference on frontiers of information technology, IEEE, 2012, pp. 257–260.
- [33] F. Thiesse, F. Michahelles, An overview of epc technology, *Sensor review* 26 (2) (2006) 101–105.
- [34] R. Minerva, A. Biru, D. Rotondi, Towards a definition of the internet of things (iot), *IEEE Internet Initiative* 1 (1) (2015) 1–86.
- [35] S. Krčo, B. Pokrić, F. Carrez, Designing iot architecture (s): A european perspective, in: 2014 IEEE world forum on internet of things (WF-IoT), IEEE, 2014, pp. 79–84.
- [36] J. Bradley, J. Loucks, A. Noronha, J. Macaulay, L. Buckalew, Internet of everything (ioe), *Survey Report* (2013).
- [37] A. Rghioui, M. Bouhorma, A. Benslimane, Analytical study of security aspects in 6lowpan networks, in: 2013 5th International Conference on Information and Communication Technology for the Muslim World (ICT4M), IEEE, 2013, pp. 1–5.
- [38] K. Zhou, T. Liu, L. Zhou, Industry 4.0: Towards future industrial opportunities and challenges, 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) (2015) 2147–2152doi:10.1109/FSKD.2015.7382284.
- [39] S. Tay, T. Lee, N. Hamid, A. Ahmad, An overview of industry 4.0: Definition, components, and government initiatives, *J. Adv. Res. Dyn. Control Syst* 10 (2018) 1379–1387.
- [40] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, *Computer networks* 54 (15) (2010) 2787–2805.
- [41] Z. Shelby, K. Hartke, C. Bormann, B. Frank, Rfc 7252: The constrained application protocol (coap), *Internet Engineering Task Force* (2014).
- [42] I. Ishaq, D. Carels, G. K. Teklemariam, J. Hoebeke, F. V. d. Abeele, E. D. Poorter, I. Moerman, P. Demeester, Ietf standardization in the field of the internet of things (iot): a survey, *Journal of Sensor and Actuator Networks* 2 (2) (2013) 235–287.

- [43] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, K. Wehrle, 6lowpan fragmentation attacks and mitigation mechanisms, in: Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks, 2013, pp. 55–66.
- [44] S. Raza, D. Tralalza, T. Voigt, 6lowpan compressed dtls for coap, in: 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems, IEEE, 2012, pp. 287–289.
- [45] M. Nawir, A. Amir, N. Yaakob, O. B. Lynn, Internet of things (iot): Taxonomy of security attacks, in: 2016 3rd International Conference on Electronic Design (ICED), IEEE, 2016, pp. 321–326.
- [46] Y. Yang, L. Wu, G. Yin, L. Li, H. Zhao, A survey on security and privacy issues in internet-of-things, IEEE Internet of Things Journal 4 (5) (2017) 1250–1258.
- [47] G. Joy Persial, M. Prabhu, R. Shanmugalakshmi, Side channel attack-survey, Int J Adva Sci Res Rev 1 (4) (2011) 54–57.
- [48] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, X. Fu, Security vulnerabilities of internet of things: A case study of the smart plug system, IEEE Internet of Things Journal 4 (6) (2017) 1899–1909.
- [49] B. A. Visan, J. Lee, B. Yang, A. H. Smith, E. T. Matson, Vulnerabilities in hub architecture iot devices, in: 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2017, pp. 83–88.
- [50] P. H. Griffin, Secure authentication on the internet of things, in: SoutheastCon 2017, IEEE, 2017, pp. 1–5.
- [51] N. Djedjig, I. Romdhani, D. Tandjaoui, F. Medjek, Trust-based defence model against mac unfairness attacks for iot, ICWMC 2017 127 (2017).
- [52] J. S. Kumar, D. R. Patel, A survey on internet of things: Security and privacy issues, International Journal of Computer Applications 90 (11) (2014).
- [53] D. Alrababah, E. Al-Shammari, A. Alsuhth, A survey: Authentication protocols for wireless sensor network in the internet of things; keys and attacks, in: 2017 International Conference on New Trends in Computing Sciences (ICTCS), IEEE, 2017, pp. 270–276.
- [54] D. M. Mendez, I. Papapanagiotou, B. Yang, Internet of things: Survey on security and privacy, arXiv preprint arXiv:1707.01879 (2017).

- [55] S. Raza, T. Voigt, V. Jutvik, Lightweight ikev2: a key management solution for both the compressed ipsec and the iee 802.15. 4 security, in: Proceedings of the IETF workshop on smart object security, Vol. 23, Citeseer, 2012.
- [56] P. Varadarajan, G. Crosby, Implementing ipsec in wireless sensor networks, in: 2014 6th International Conference on New Technologies, Mobility and Security (NTMS), IEEE, 2014, pp. 1–5.
- [57] A. Rghioui, A. Khannous, M. Bouhorma, Denial-of-service attacks on 6lowpan-rpl networks: Threats and an intrusion detection system proposition, *Journal of Advanced Computer Science & Technology* 3 (2) (2014) 143–153.
- [58] S. Raza, L. Wallgren, T. Voigt, Svelte: Real-time intrusion detection in the internet of things, *Ad hoc networks* 11 (8) (2013) 2661–2674.
- [59] J. Granjal, E. Monteiro, J. S. Silva, Security for the internet of things: a survey of existing protocols and open research issues, *IEEE Communications Surveys & Tutorials* 17 (3) (2015) 1294–1312.
- [60] A. A. Ghorbani, W. Lu, M. Tavallaee, Network intrusion detection and prevention: concepts and techniques, Vol. 47, Springer Science & Business Media, 2009.
- [61] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of machine learning, MIT press, 2018.
- [62] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, *Nature* 521 (2015) 436–444.
- [63] S. R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, *IEEE transactions on systems, man, and cybernetics* 21 (3) (1991) 660–674. doi:10.1109/21.97458.
- [64] L. Breiman, Random forests, *Machine learning* 45 (1) (2001) 5–32.
- [65] J. M. Keller, M. R. Gray, J. A. Givens, A fuzzy k-nearest neighbor algorithm, *IEEE transactions on systems, man, and cybernetics* (4) (1985) 580–585. doi:10.1109/TSMC.1985.6313426.
- [66] T. M. Mitchell, et al., *Machine learning*. 1997, Burr Ridge, IL: McGraw Hill 45 (37) (1997) 870–877.
- [67] M. Riedmiller, Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms, *Computer Standards & Interfaces* 16 (3) (1994) 265–278.

- [68] D. W. Hosmer Jr, S. Lemeshow, R. X. Sturdivant, Applied logistic regression, Vol. 398, John Wiley & Sons, Ltd, 2013. doi:10.1002/9781118548387.
- [69] A. Zappone, M. D. Renzo, M. Debbah, Wireless networks design in the era of deep learning: Model-based, ai-based, or both?, IEEE Transactions on Communications 67 (2019) 7331–7376.
- [70] L. Deng, A tutorial survey of architectures, algorithms, and applications for deep learning, APSIPA Transactions on Signal and Information Processing 3 (2014).
- [71] J. Vasseur, M. Kim, K. Pister, N. Dejean, D. Barthel, Routing metrics used for path calculation in low power and lossy networks, RFC 6551, Internet Engineering Task Force (2012).
- [72] P. Thubert, Objective function zero for the routing protocol for low-power and lossy networks (rpl), RFC 6552, Internet Engineering Task Force (2012).
- [73] O. Gnawali, P. Levis, The minimum rank with hysteresis objective function, RFC 6719 (2012).
- [74] T. Clausen, U. Herberg, M. Philipp, A critical evaluation of the ipv6 routing protocol for low power and lossy networks (rpl), in: 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2011, pp. 365–372.
- [75] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, The trickle algorithm, Internet Engineering Task Force, RFC6206 (2011).
- [76] A. Dunkels, B. Gronvall, T. Voigt, Contiki-a lightweight and flexible operating system for tiny networked sensors, in: 29th annual IEEE international conference on local computer networks, IEEE, 2004, pp. 455–462.
- [77] Contiki-os [online].  
URL <https://github.com/contiki-os/contiki/tree/master/core/net/rpl>
- [78] N. Tsiftes, J. Eriksson, A. Dunkels, Low-power wireless ipv6 routing with contikirpl, in: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2010, pp. 406–407.
- [79] Tinyos [online].  
URL <https://github.com/tinyos/tinyos-main/tree/master/tos/lib/net/rpl>

- [80] Riot-os [online].  
URL <https://github.com/RIOT-OS/RIOT/tree/master/sys/net/gnrc/routing/rpl>
- [81] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, T. C. Schmidt, Riot os: Towards an os for the internet of things, in: 2013 IEEE conference on computer communications workshops (INFOCOM WKSHPs), IEEE, 2013, pp. 79–80.
- [82] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-level sensor network simulation with cooja, in: Proceedings. 2006 31st IEEE Conference on Local Computer Networks, 2006, pp. 641–648. doi:10.1109/LCN.2006.322172.
- [83] D. Airehrour, J. Gutierrez, S. K. Ray, Secure routing for internet of things: A survey, *Journal of Network and Computer Applications* 66 (2016) 198–213.
- [84] A. Le, J. Loo, Y. Luo, A. Lasebae, Specification-based ids for securing rpl from topology attacks, in: *Wireless Days (WD)*, 2011 IFIP, IEEE, 2011, pp. 1–3.
- [85] A. Le, J. Loo, A. Lasebae, M. Aiash, Y. Luo, 6lowpan: a study on qos security threats and countermeasures using intrusion detection system approach, *International Journal of Communication Systems* 25 (9) (2012) 1189–1212.
- [86] A. Le, J. Loo, Y. Luo, A. Lasebae, The impacts of internal threats towards routing protocol for low power and lossy network performance, in: *Computers and Communications (ISCC)*, 2013 IEEE Symposium on, IEEE, 2013, pp. 000789–000794.
- [87] A. Dvir, T. Holczer, L. Buttyan, Vera-version number and rank authentication in rpl, in: *Mobile Adhoc and Sensor Systems (MASS)*, 2011 IEEE 8th International Conference on, IEEE, 2011, pp. 709–714.
- [88] U. Shafique, A. Khan, A. Rehman, F. Bashir, M. Alam, Detection of rank attack in routing protocol for low power and lossy networks, *Annals of Telecommunications* 73 (7) (2018) 429–438.
- [89] A. Yahyaoui, F. Yaakoubi, T. Abdellatif, et al., Machine learning based rank attack detection for smart hospital infrastructure, in: *International Conference on Smart Homes and Health Telematics*, Springer, 2020, pp. 28–40.
- [90] A. Rehman, M. M. Khan, M. A. Lodhi, F. B. Hussain, Rank attack using objective function in rpl for low power and lossy networks, in: 2016 International

Conference on Industrial Informatics and Computer Systems (CIICS), IEEE, 2016, pp. 1–5.

- [91] L. Zhang, G. Feng, S. Qin, Intrusion detection system for rpl from routing choice intrusion, in: 2015 IEEE International Conference on Communication Workshop (ICCW), IEEE, 2015, pp. 2652–2658.
- [92] J. Hui, J. Vasseur, Rpl option for carrying rpl information in data-plane datagrams, draft-ietf-6man-rpl-option-03 (work in progress) (2011).
- [93] B. Ghaleb, A. Al-Dubai, E. Ekonomou, M. Qasem, I. Romdhani, L. Mackenzie, Addressing the dao insider attack in rpl’s internet of things networks, IEEE Communications Letters 23 (1) (2019) 68–71. doi:10.1109/LCOMM.2018.2878151.
- [94] I. Wadhaj, B. Ghaleb, C. Thomson, A. Al-Dubai, W. J. Buchanan, Mitigation mechanisms against the dao attack on the routing protocol for low power and lossy networks (rpl), IEEE Access 8 (2020) 43665–43675.
- [95] A. S. Baghani, S. Rahimpour, M. Khabbazian, The dao induction attack against the rpl-based internet of things, in: 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2020, pp. 1–5. doi:10.23919/SoftCOM50211.2020.9238224.
- [96] C. Pu, Spam dis attack against routing protocol in the internet of things, in: 2019 International Conference on Computing, Networking and Communications (ICNC), 2019, pp. 73–77. doi:10.1109/ICNC.2019.8685628.
- [97] A. Verma, V. Ranga, Mitigation of dis flooding attacks in rpl-based 6lowpan networks, Trans. Emerg. Telecommun. Technol. 31 (2020).
- [98] F. Medjek, D. Tandjaoui, N. Djedjig, I. Romdhani, Fault-tolerant ai-driven intrusion detection system for the internet of things, International Journal of Critical Infrastructure Protection 34 (2021) 100436.
- [99] R. Smith, D. Palin, P. P. Ioulianou, V. G. Vassilakis, S. Shahandashti, Battery draining attacks against edge computing nodes in iot networks, Cyber-Physical Systems 6 (2020) 116 – 96.
- [100] S. Sharma, V. K. Verma, Security explorations for routing attacks in low power networks on internet of things, The Journal of Supercomputing (2020) 1–35.
- [101] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, J. Schönwälder, A study of rpl dodag version attacks, in: IFIP international conference on autonomous infrastructure, management and security, Springer, 2014, pp. 92–104.



- [102] A. Aris, S. F. Oktug, S. B. O. Yalcin, Rpl version number attacks: In-depth study, in: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2016, pp. 776–779.
- [103] L. Wallgren, S. Raza, T. Voigt, Routing attacks and countermeasures in the rpl-based internet of things, *International Journal of Distributed Sensor Networks* 2013 (2013).
- [104] F. Medjek, D. Tandjaoui, N. Djedjig, I. Romdhani, Multicast dis attack mitigation in rpl-based iot-llns, *Journal of Information Security and Applications* 61 (2021) 102939.
- [105] Z. Haas, L. Yang, M. Liu, Q. Li, F. Li, Current challenges and approaches in securing communications for sensors and actuators, in: *The Art of Wireless Sensor Networks*, Springer, 2014, pp. 569–608.
- [106] K. Weekly, K. Pister, Evaluating sinkhole defense techniques in rpl networks, in: *2012 20th IEEE International Conference on Network Protocols (ICNP)*, IEEE, 2012, pp. 1–6.
- [107] A. Kumar, R. Matam, S. Shukla, Impact of packet dropping attacks on rpl, in: *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, IEEE, 2016, pp. 694–698.
- [108] D. Airehrour, J. Gutierrez, S. K. Ray, Securing rpl routing protocol from blackhole attacks using a trust-based mechanism, in: *2016 26th International Telecommunication Networks and Applications Conference (ITNAC)*, IEEE, 2016, pp. 115–120.
- [109] E. Shi, A. Perrig, Designing secure sensor networks, *IEEE Wireless Communications* 11 (6) (2004) 38–43.
- [110] T. Kavitha, D. Sridharan, Security vulnerabilities in wireless sensor networks: A survey, *Journal of information Assurance and Security* 5 (1) (2010) 31–44.
- [111] J. R. Renofio, M. E. Pellenz, E. Jamhour, A. Santin, M. C. Penna, R. D. Souza, On the dynamics of the rpl protocol in ami networks under jamming attacks, in: *2016 IEEE International Conference on Communications (ICC)*, IEEE, 2016, pp. 1–6.
- [112] P. Perazzo, C. Vallati, A. Arena, G. Anastasi, G. Dini, An implementation and evaluation of the security features of rpl, in: *International Conference on Ad-Hoc Networks and Wireless*, Springer, 2017, pp. 63–76.

- [113] M. R. Abdmeziem, D. Tandjaoui, I. Romdhani, Lightweighted and energy-aware mikey-ticket for e-health applications in the context of internet of things, *International Journal of Sensor Networks* 26 (4) (2018) 227–242.
- [114] N. Djedjig, D. Tandjaoui, F. Medjek, Trust-based rpl for the internet of things, in: *2015 IEEE Symposium on Computers and Communication (ISCC)*, IEEE, 2015, pp. 962–967.
- [115] N. Djedjig, D. Tandjaoui, F. Medjek, I. Romdhani, New trust metric for the rpl routing protocol, in: *Information and Communication Systems (ICICS)*, 2017 8th International Conference on, IEEE, 2017, pp. 328–335.
- [116] N. Djedjig, D. Tandjaoui, F. Medjek, I. Romdhani, Trust-aware and cooperative routing protocol for iot security, *Journal of Information Security and Applications* 52 (2020) 102467.
- [117] Z. A. Khan, J. Ullrich, A. G. Voyiatzis, P. Herrmann, A trust-based resilient routing mechanism for the internet of things, in: *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ACM, 2017, p. 27.
- [118] A. Le, J. Loo, K. K. Chai, M. Aiash, A specification-based ids for detecting attacks on rpl-based network topology, *Information* 7 (2) (2016) 25.
- [119] M. Surendar, A. Umamakeswari, Indres: An intrusion detection and response system for internet of things with 6lowpan, in: *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, IEEE, 2016, pp. 1903–1908.
- [120] C. Liu, J. Yang, R. Chen, Y. Zhang, J. Zeng, Research on immunity-based intrusion detection technology for the internet of things, in: *2011 Seventh International Conference on Natural Computation*, Vol. 1, IEEE, 2011, pp. 212–216.
- [121] P. Kasinathan, C. Pastrone, M. A. Spirito, M. Vinkovits, Denial-of-service detection in 6lowpan based internet of things, in: *2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob)*, IEEE, 2013, pp. 600–607.
- [122] D. Oh, D. Kim, W. W. Ro, A malicious pattern detection engine for embedded security systems in the internet of things, *Sensors* 14 (12) (2014) 24188–24211.

- [123] P. Pongle, G. Chavan, Real time intrusion and wormhole attack detection in internet of things, *International Journal of Computer Applications* 121 (9) (2015).
- [124] C. Cervantes, D. Poplade, M. Nogueira, A. Santos, Detection of sinkhole attacks for supporting secure routing on 6lowpan for internet of things, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, 2015, pp. 606–611.
- [125] N. K. Thanigaivelan, E. Nigussie, S. Virtanen, J. Isoaho, Hybrid internal anomaly detection system for iot: Reactive nodes with cross-layer operation, *Security and Communication Networks* 2018 (2018).
- [126] F. Gara, L. B. Saad, R. B. Ayed, An intrusion detection system for selective forwarding attack in ipv6-based mobile wsns, in: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2017, pp. 276–281.
- [127] P. P. Ioulianou, V. G. Vassilakis, Denial-of-service attacks and countermeasures in the rpl-based internet of things, in: S. Katsikas, F. Cuppens, N. Cuppens, C. Lambrinouidakis, C. Kalloniatis, J. Mylopoulos, A. Antón, S. Gritzalis, F. Pallas, J. Pohle, A. Sasse, W. Meng, S. Furnell, J. Garcia-Alfaro (Eds.), *Computer Security*, Springer International Publishing, 2020, pp. 374–390.
- [128] B. Farzaneh, M. A. Montazeri, S. Jamali, An anomaly-based ids for detecting attacks in rpl-based internet of things, in: 2019 5th International Conference on Web Research (ICWR), 2019, pp. 61–66. doi:10.1109/ICWR.2019.8765272.
- [129] M. Sheikhan, H. Bostani, A security mechanism for detecting intrusions in internet of things using selected features based on mi-bgsa, *International Journal of Information and Computer Technology Research* 9 (2) (2017) 53–62.
- [130] C. D. McDermott, A. Petrovski, Investigation of computational intelligence techniques for intrusion detection in wireless sensor networks, *International Journal of Computer Networks and Communications* 9 (4) (2017).
- [131] M. N. Napiah, M. Y. I. B. Idris, R. Ramli, I. Ahmedy, Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol, *IEEE Access* 6 (2018) 16623–16638. doi:10.1109/ACCESS.2018.2798626.

- [132] E. Anthi, L. Williams, P. Burnap, Pulse: an adaptive intrusion detection for the internet of things, in: *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, IET, 2018, pp. 1–4. doi:10.1049/cp.2018.0035.
- [133] M. Hasan, M. M. Islam, M. I. I. Zarif, M. Hashem, Attack and anomaly detection in iot sensors in iot sites using machine learning approaches, *Internet of Things* 7 (2019) 100059. doi:10.1016/j.iot.2019.100059.
- [134] R. Primartha, B. A. Tama, Anomaly detection using random forest: A performance revisited, *2017 International Conference on Data and Software Engineering (ICoDSE)* (2017) 1–6.
- [135] E. Min, J. Long, Q. Liu, J. Cui, W. Chen, Tr-ids: Anomaly-based intrusion detection through text-convolutional neural network and random forest, *Secur. Commun. Networks* 2018 (2018) 4943509:1–4943509:9.
- [136] B. A. Tama, K.-H. Rhee, An integration of pso-based feature selection and random forest for anomaly detection in iot network, 2018.
- [137] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, H. Ming, Ad-iot: Anomaly detection of iot cyberattacks in smart city using machine learning, *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)* (2019) 0305–0310.
- [138] A. A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for internet of things, *Future Generation Computer Systems* 82 (2018) 761 – 768. doi:10.1016/j.future.2017.08.043.
- [139] F. Y. Yavuz, D. Ünal, E. Gül, Deep learning for detection of routing attacks in the internet of things, *International Journal of Computational Intelligence Systems* 12 (1) (2018) 39–58. doi:10.2991/ijcis.2018.25905181.
- [140] A.-U.-H. Qureshi, H. Larijani, J. Ahmad, N. Mtetwa, A heuristic intrusion detection system for internet-of-things (iot), 2019.
- [141] J. R. Douceur, The sybil attack, in: *International Workshop on Peer-to-Peer Systems*, Springer, 2002, pp. 251–260.
- [142] J. Newsome, E. Shi, D. Song, A. Perrig, The sybil attack in sensor networks: analysis & defenses, in: *Proceedings of the 3rd international symposium on Information processing in sensor networks*, ACM, 2004, pp. 259–268.
- [143] K. Zhang, X. Liang, R. Lu, X. Shen, Sybil attacks and their defenses in the internet of things, *IEEE Internet of Things Journal* 1 (5) (2014) 372–383.

- [144] S. Pawar, P. Vanwari, Sybil attack in internet of things, *International Journal of Engineering and Innovative Technology (IJESIT)* (2016).
- [145] T. Silawan, C. Aswakul, Sybilcomm: Sybil community detection using persuading function in iot system, in: *2016 International Conference on Electronics, Information, and Communications (ICEIC)*, IEEE, 2016, pp. 1–4.
- [146] D. Evangelista, F. Mezghani, M. Nogueira, A. Santos, Evaluation of sybil attack detection approaches in the internet of things content dissemination, in: *Wireless Days (WD)*, 2016, IEEE, 2016, pp. 1–6.
- [147] S. Thomson, Ipv6 stateless address autoconfiguration, RFC 4862 (1998).
- [148] T. Preiss, M. Sherburne, R. Marchany, J. Tront, Implementing dynamic address changes in contikios, in: *2014 International Conference on Information Society (i-Society)*, IEEE, 2014, pp. 222–227.
- [149] R. Vida, L. Costa, Rfc 3810, Multicast Listener Discovery Version 2 (2004).
- [150] P. Thubert, Ipv6 backbone router draft-ietf-6lo-backbone-router-03 (2017).
- [151] M. Sethi, P. Thubert, B. Sarikaya, Address protected neighbor discovery for low-power and lossy networks draft-sarikaya-6lo-ap-nd-04 (2016).
- [152] S. Choudhary, N. Kesswani, A survey: Intrusion detection techniques for internet of things, *International Journal of Information Security and Privacy (IJISP)* 13 (1) (2019) 86–105.
- [153] M. Hossin, M. Sulaiman, A review on evaluation metrics for data classification evaluations, *International Journal of Data Mining & Knowledge Management Process* 5 (2) (2015) 1. doi:10.5121/ijdkp.2015.5201.
- [154] J. Krawczuk, T. Łukaszuk, The feature selection bias problem in relation to high-dimensional gene data, *Artificial intelligence in medicine* 66 (2016) 63–71. doi:10.1016/j.artmed.2015.11.001.
- [155] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, *Journal of machine learning research* 5 (Oct) (2004) 1205–1224.
- [156] G. E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, *Neural computation* 18 (7) (2006) 1527–1554.

## Abstract

The Internet of Things (IoT) consists of physical objects that sense, collect, and might process data. These objects are resource-constrained as they are powered by batteries and have limited computation and storage capability. Billions of these devices are interconnected and connected to the Internet under lossy and noisy communication environments such as ZigBee and Bluetooth. IoT applications have emerged in several socio-economic sectors such as healthcare, industry, and energy. Nevertheless, the IoT's Low-Power and Lossy Networks (LLNs) rise challenges in designing efficient and secure routing protocols that fulfil the routing requirements in such networks. In this regards, the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) was designed and standardised to overcome the routing challenges underpinning the LLNs. RPL considers limitations in both the energy power and the computational capabilities of LLNs. Besides the different characteristics of the IoT components, the rapid growth of IoT applications and the increasing number of smart objects result in producing a massive amount of data and traffic leading to increase the IoT's vulnerabilities. The research community studies the security challenges of the IoT from many different points of view, one of which is the security vulnerability of IoT communication protocols at the network layer, and consequently, the RPL's threats given that it represents one of the main pillars of LLNs. Although the RPL specification introduces mechanisms aiming to achieve confidentiality, integrity and replay protection, RPL is still susceptible to internal attacks that go beyond the encryption and authentication defence. In response to the RPL's security issues, intrusions tolerance and detection systems are proposed in this work as the last line of defence for RPL. On the one hand, intrusion detection systems (IDSs) have been proposed to analyse the network's activities and nodes' behaviour in order to detect the intrusions. On the other hand, intrusion tolerance mechanisms have been introduced to respond immediately to the intrusions, thus reducing the effects of the attacks on the LLNs.

**Keywords:** IoT, LNN, RPL, RPL security, Intrusion Tolerance, Intrusion Detection Systems, Machine Learning.

## Résumé

L'Internet des objets (IoT) se compose d'objets physiques qui détectent, collectent et peuvent traiter des données. Ces objets sont limités en ressources car ils sont alimentés par des batteries et ont une capacité de calcul et de stockage limitée. Des milliards de ces objets sont interconnectés et connectés à Internet dans des environnements de communication bruyants et avec perte tels que ZigBee et Bluetooth. Les applications IoT ont émergé dans plusieurs secteurs socio-économiques tels que la santé, l'industrie et l'énergie. Néanmoins, les réseaux à faible consommation et avec perte (LLN) de l'IoT posent des défis dans la conception de protocoles de routage efficaces et sécurisés qui répondent aux exigences de routage de ces réseaux. À cet égard, le protocole de routage IPv6 pour les réseaux à faible consommation et avec perte (RPL) a été conçu et normalisé pour surmonter les défis de routage qui sous-tendent les LLN. Le RPL prend en compte les limites à la fois de la puissance énergétique et des capacités de calcul des LLN. Outre les différentes caractéristiques des composants IoT, la croissance rapide des applications IoT et le nombre croissant d'objets intelligents entraînent la production d'une quantité massive de données et de trafic, ce qui augmente les vulnérabilités de l'IoT. La communauté de recherche étudie les défis de sécurité de l'IoT sous de nombreux points de vue, dont l'un est la vulnérabilité des protocoles de communication IoT au niveau de la couche réseau, et par conséquent, les menaces de sécurité du RPL étant donné qu'il représente l'un des principaux piliers des LLN. Bien que la spécification RPL introduise des mécanismes visant à assurer la confidentialité, l'intégrité et la protection contre la relecture, RPL est toujours sensible aux attaques internes qui vont au-delà du cryptage et de la défense d'authentification. Par conséquent, en réponse à de tels problèmes de sécurité du RPL, des systèmes de tolérance et de détection d'intrusions sont proposés dans ce travail comme dernière ligne de défense du RPL. D'une part, des systèmes de détection d'intrusion (IDS) ont été proposés pour analyser les activités du réseau et le comportement des nœuds afin de détecter les intrusions. D'autre part, des mécanismes de tolérance aux intrusions ont été introduits pour répondre immédiatement aux intrusions, réduisant ainsi les effets des attaques sur les LLN.

**Mots-clés :** IoT, LNN, RPL, sécurité RPL, tolérance aux intrusions, systèmes de détection d'intrusions, apprentissage automatique.

## الملخص

تتكون إنترنت الأشياء (IoT) من أشياء مادية تستشعر البيانات وتجمعها وقد تعالجها. هذه الأشياء محدودة الموارد لأنها تعمل بالبطاريات ولديها قدرة محدودة على الحساب والتخزين. المليارات من هذه الأجهزة متصلة ببعضها البعض ومتصلة بالإنترنت في ظل بيئات اتصال ضائعة وصاخبة مثل ZigBee و Bluetooth. ظهرت تطبيقات إنترنت الأشياء في العديد من القطاعات الاجتماعية والاقتصادية مثل الرعاية الصحية والصناعة والطاقة. ومع ذلك فإن شبكات إنترنت الأشياء منخفضة الطاقة والمفقودة (LLNs) تزيد من التحديات في تصميم بروتوكولات توجيه فعالة وأمنة تلبي متطلبات التوجيه في مثل هذه الشبكات. في هذا الصدد، تم تصميم وتوحيد بروتوكول توجيه IPv6 للشبكات منخفضة الطاقة وفقدان (RPL) للتغلب على تحديات التوجيه التي تدعم شبكات LLN. يأخذ RPL في الاعتبار القيود في كل من قوة الطاقة والقدرات الحسابية لـ LLNs. إلى جانب الخصائص المختلفة لمكونات إنترنت الأشياء، يؤدي النمو السريع لتطبيقات إنترنت الأشياء والعدد المتزايد من الأجهزة الذكية إلى إنتاج كمية هائلة من البيانات وحركة المرور مما يؤدي إلى زيادة نقاط ضعف إنترنت الأشياء. يدرس المجتمع البحثي التحديات الأمنية لإنترنت الأشياء من عدة وجهات نظر، أحدها هو الضعف الأمني لبوتوكولات اتصالات إنترنت الأشياء في طبقة الشبكة، وبالتالي التهديدات الأمنية على RPL نظرًا لأنها تمثل أحد الركائز الأساسية لـ LLNs. على الرغم من أن مواصفات RPL تقدم آليات تهدف إلى تحقيق السرية والنزاهة وحماية إعادة التشغيل، إلا أن RPL لا يزال عرضة للهجمات الداخلية التي تتجاوز دفاع التشفير والمصادقة. استجابةً لقضايا الأمن الخاصة بـ RPL، تم في هذا العمل اقتراح أنظمة تحمل الاختراقات والكشف عنها كخط دفاع أخير لـ RPL. من ناحية، تم اقتراح أنظمة كشف الاختراقات (IDS) لتحليل أنشطة الشبكة وسلوك العقد من أجل الكشف عن عمليات الاختراقات. من ناحية أخرى، تم إدخال آليات تحمل الاختراقات للرد الفوري على عمليات الاقتحام، وبالتالي تقليل آثار الهجمات على LLNs.

**الكلمات الرئيسية:** IoT، LNN، RPL، أمن RPL، تحمل الاختراقات، أنظمة كشف الاختراقات، التعلم الآلي.