Intrusion Detection System for Detecting Internal Threats in 6LoWPAN

By

Anhtuan Le

Student number: M00175252

Supervisors: Dr Jonathan Loo

Dr Aboubaker Lasebae

Dr Yuan Luo

Thesis submitted to the Middlesex University for the degree of Doctor of Philosophy

London, February 2016

Abstract

6LoWPAN (IPv6 over Low-power Wireless Personal Area Network) is a standard developed by the Internet Engineering Task Force group to enable the Wireless Sensor Networks to connect to the IPv6 Internet. This standard is rapidly gaining popularity for its applicability, ranging extensively from health care to environmental monitoring. Security is one of the most crucial issues that need to be considered properly in 6LoWPAN. Common 6LoWPAN security threats can come from external or internal attackers. Cryptographic techniques are helpful in protecting the external attackers from illegally joining the network. However, because the network devices are commonly not tampered-proof, the attackers can break the cryptography codes of such devices and use them to operate like an internal source. These malicious sources can create internal attacks, which may downgrade significantly network performance. Protecting the network from these internal threats has therefore become one of the centre security problems on 6LoWPAN.

This thesis investigates the security issues created by the internal threats in 6LoWPAN and proposes the use of Intrusion Detection System (IDS) to deal with such threats. Our main works are to categorise the 6LoWPAN threats into two major types, and to develop two different IDSs to detect each of this type effectively. The major contributions of this thesis are summarised as below.

First, we categorise the 6LoWPAN internal threats into two main types, one that focuses on compromising directly the network performance (performance-type) and the other is to manipulate the optimal topology (topology-type), to later downgrade the network service quality indirectly. In each type, we select some typical threats to implement, and assess their particular impacts on network performance as well as identify performance metrics that are sensitive in the attacked situations, in order to form the basis detection knowledge. In addition, on studying the topology-type, we propose several novel attacks towards the Routing Protocol for Low Power and Lossy network (RPL - the underlying routing protocol in 6LoWPAN), including the Rank attack, Local Repair attack and DIS attack.

Second, we develop a Bayesian-based IDS to detect the performance-type internal threats by monitoring typical attacking targets such as traffic, channel or neighbour nodes. Unlike other statistical approaches, which have a limited view by just using a single metric to monitor a specific attack, our Bayesian-based IDS can judge an abnormal behaviour with a wiser view by considering of different metrics using the insightful understanding of their relations. Such wiser view helps to increase the IDS's accuracy significantly.

Third, we develop a Specification-based IDS module to detect the topology-type internal threats based on profiling the RPL operation. In detail, we generalise the observed states and transitions of RPL control messages to construct a high-level abstract of node operations through analysing the trace files of the simulations. Our profiling technique can form all of the protocol's legal states and transitions automatically with corresponding statistic data, which is faster and easier to verify compare with other manual specification techniques. This IDS module can detect the topology-type threats quickly with a low rate of false detection.

We also propose a monitoring architecture that uses techniques from modern technologies such as LTE (Long-term Evolution), cloud computing, and multiple interface sensor devices, to expand significantly the capability of the IDS in 6LoWPAN. This architecture can enable the running of both two proposed IDSs without much overhead created, to help the system to deal with most of the typical 6LoWPAN internal threats.

Overall, the simulation results in Contiki Cooja prove that our two IDS modules are effective in detecting the 6LoWPAN internal threats, with the detection accuracy is ranging between 86 to 100% depends on the types of attacks, while the False Positive is also satisfactory, with under 5% for most of the attacks. We also show that the additional energy consumptions and the overhead of the solutions are at an acceptable level to be used in the 6LoWPAN environment.

Acknowledgements

I would like to use this opportunity to acknowledge my debt of gratitude to my supervisors, colleagues, friends and family.

First, I would like to express my deepest gratitude to my Director of Study, my mentor, Dr Jonathan Loo, for his continuous guidance, kind supports and great enthusiasm for high-quality research. Dr Loo has given me inspiration in virtually everything I have done at Middlesex University. I have benefited greatly from his invaluable advice on both my research and life. Without him, it was definitely impossible for me to overcome the crisis I had during the last two years of my PhD.

I also appreciate Dr Aboubaker Lasebae and Dr Yuan Luo for their guidance during my study, back from the time when I was an MSc student just coming to London. I am grateful to all my friends from TG21 who have shared the moments of happiness and disappointment with me throughout the years I have spent at Middlesex University.

Lastly, I wish to thank my family for their understanding, continued love and support during time.

Table of content

List of Figures

List of Tables

List of Abbreviations

Acronym	Definition
6LoWPAN	IPv6 over Low-power Wireless Personal Area Network
ACK	Acknowledgement
AES	Advance Encryption Standard
AODV	Ad hoc On-Demand Distance Vector Routing Protocol
BN	Bayesian Network
CPT	Conditional Probability Table
CTS	Clear To Send
DAG	Directed Acyclic Graph
DAO	Destination Advertisement Object
DIO	DODAG Information Objective
DIS	DODAG Information Solicitation
DODAG	Destination Oriented DAG
DSDV	Destination-Sequenced Distance-Vector Routing Protocol
DSR	Dynamic Source Routing
ECC	Elliptic Curve Cryptography
EFSM	Extended Finite State Machines
ESP	Encapsulation Security Payload
ETX	Expected Transmission Count
FPR	False Positive Rate
HMM	Hidden Markov Model
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
LBR	Local Border Router
MAC	Media Access Control
MAP	Maximum A Posteriori
OCP	Objective Code Point
OLSR	Optimised Link State Routing Protocol

OSPF	Open Shortest Path First routing protocol
RFC	Request for Comments
RFID	Radio-frequency identification
RIP	Routing Information Protocol
ROLL	Routing Over Low power and Lossy networks
RPL	Routing Protocol for Low-Power and Lossy Networks
RREQ	Route Request Packet
RRPL	Route Reply Packet
RSA	Rivest-Shamir-Adleman public-key cryptosystem
RSSI	Received Signal Strength Indicator
RTS	Request To Send
SVM	Support Vector Machine
TCP	Transmission Control Protocol

List of Publication from PhD research

A. Le, J. Loo, K. K. Chai, and M. Aiash, "**A Specification-Based IDS for Detecting Attacks on RPL-Based Network Topology**," *Information*, vol. 7, p. 25, 2016.

A. Le, J. Loo, A. Lasebae, A. Vinel, C. Yue, and M. Chai, "**The Impact of Rank Attack on Network Topology of Routing Protocol for Low-Power and Lossy Networks**," *Sensors Journal*, IEEE, vol. 13, pp. 3685-3692, 2013.

A. Le, J. Loo, L. Yuan, and 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, 2013, pp. 000789-000794.

A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo, "**6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach**," *International Journal of Communication Systems*, vol. 25, pp. 1189-1212, 2012.

A. Le, J. Loo, L. Yuan, and A. Lasebae, "**Specification-based IDS for securing RPL from topology attacks**," in *Wireless Days (WD)*, 2011 IFIP, 2011, pp. 1-3.

A. Le, J. Loo, et. al. "**A Bayesian-based IDS for detecting internal threats on 6LoWPAN**," submitted on *Special Issue in Internet of Things*, Elsevier Journal

A. Le, J. Loo, et. al. "**A study of sensitive network metrics under 6LoWPAN internal attacks**" to be submitted on Internet of Things Journal, IEEE

CHAPTER 1. INTRODUCTION

1.1. Background

Over the past two decades, the Internet has grown from a small academic network into a global, ubiquitous network reaching more than one third of the world's population. This growth will be continued in the future, given the rapid development of technology, both in hardware and software. Over the last few decades, sensor devices with the ability of sensing and gathering different kinds of information are predicted to be the centre of technology development. People have dreamed of a world with billions of sensor devices to monitor and control important things. Connected sensors can give humans unlimited access to the information they want. Many new applications will become more feasible which will evolve human life. This idea is the main content of the "Internet of Things" (IoT) concept, with 6LoWPAN as a fundamental technical standard to bring it to real life.

1.1.1. The Internet of Things

IoT has been developed rapidly in recent years while attracting the momentum from both the academia and the industry. Its main idea is to utilize the standard Internet protocols to interconnect smart objects for harvesting data and information. Such devices can bring various functions like identification, location, tracking, monitoring, etc. [1, 2]. With the fast development in Radio Frequency Identification (RFID), embedded sensors, miniature actuators, nanotechnology [3], IoT now has a wide range of real life applications, from transportation and logistics, health care, smart environment, to personal and social, gaming, robot, city information (see illustration in Figure 1.1) [3]. With IoT, the Information and Communication Technology (ICT) is expected to see a paradigm shift from human-to-human communication style to human-to-thing, and thing-to-thing [3]. Smart objects can connect, exchange information and even make decisions on behalf of users. This will bring a new connectivity dimension, which provides connectivity for anything, for anyone at anytime and anyplace [4].

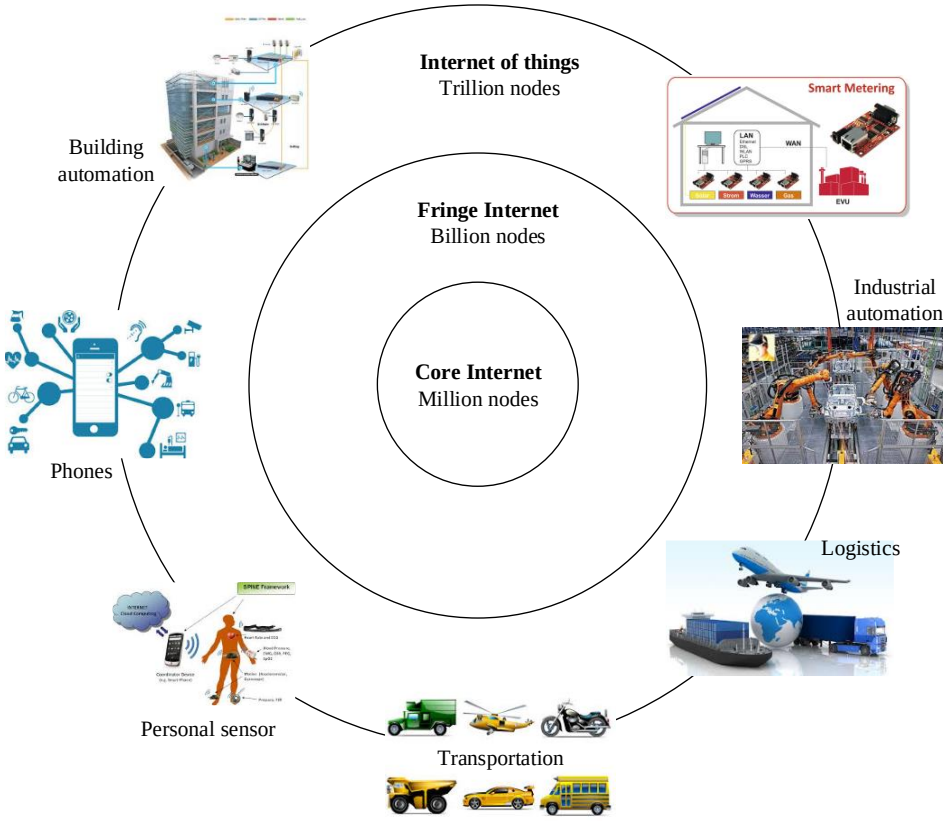


Figure 1.1. IoT applications in real life

1.1.2. 6LoWPAN

The IoT devices have specific characteristics such as short radio range, limited processing capability and short battery life. Therefore, implementation of the concept requires a communication framework that can efficiently manage these resource constrains. 6LoWPAN, a proposed standard that utilize IPv6 to connect the IoT devices through an adaptation layer, is a promising solution. This standard can allow the use of the existing IP infrastructure to maximise the utilisation of available resources, while benefiting from the huge address space of IPv6. Moreover, the implementation can be accelerated by using previous tools and mechanisms to save time and efforts in development.

1.1.3. Security Issues in 6LoWPAN

Due to its open architecture [5], 6LoWPAN security problems need to be considered carefully for the standard to be publicly deployable. On implementing 6LoWPAN, most of the security threats, which come from 802.15.4, IP network and its adaptation layer, become more specific. The 802.15.4 part has weaker security than the IPv6 part. Its resource-constrained devices are usually not tampered-proof so attackers can tamper and

take control of these devices (see [6]). Once the attackers join the network, they can have many different ways to downgrade the performance; for instance, dropping the packets that need to be followed, or sending extra packets to neighbours for interrupting their operations. Moreover, the distributed manner of this side makes it have limited support for security services. Security threats for this part can come from both the external and internal attackers and target all the layers. Threats on the IP part, on the other hand, are mostly related to user authentication and data integrity. For example, unauthenticated users can access the information on the LoWPAN part, or falsify the data sending from the sensor. Furthermore, the adaptation layer to connect the two parts, is also vulnerable, for instance, the fragmentation attacks that disrupt the operation [7].

1.2. Motivation

Many 6LoWPAN applications will strictly require the information security and robustness quality of services. Cryptography can be used as the first line of defense to satisfy such requirements. However, the sensor devices are normally cheap and non tamper-resistant, so the adversaries can bypass the cryptographic to join the network as a legal node. Once this can be done, attacker can change the operation of the compromised nodes to launch different attacks. For instance, such nodes can cause the disruption of route discovery or data forwarding; or modify packet contents from legitimate nodes to form severe routing attacks. The purposes of such actions are mainly to downgrade the network QoS to disturb the applications. These so-called internal threats are difficult to be detected because all malicious nodes are still legal as long as they are authorised by the keys.

Our main motivation is to secure the 6LoWPAN performance from such internal threats. In order to do so, we study the behaviours of the internal attackers, and apply the understanding of the attacks in the monitoring systems to detect them. The work in this thesis has started since 2010. At that time, there was only a little research about internal threats in 6LoWPAN. Through time, this topic is getting more and more attention from both the academic and industry, which has proved our foreseen about the importance of the topic.

1.3. Research Questions

In this thesis, we aim to answer the following research questions:

1. *What are the typical internal threats in 6LoWPAN? What are the potential impacts on network performance, and how to measure them? How do internal attackers behave?*
2. *How to build an Intrusion Detection System(s) to detect internal attackers? The system(s) should have a high accuracy of detection, and should be energy-efficiency to operate within 6LoWPAN resource constraint.*

Through thoroughly analysing and quantifying 6LoWPAN internal threats, we have realised internal attacks can be roughly divided into two types:

- Those who aim to downgrade directly the network performance (performance-type), targeting mainly traffic, channel, and neighbour nodes.
- Those who aim to break the optimal topology (topology-type), which target protocol operation.

Our work employs selected network metrics to study the attack impacts and node behaviours. The results of such study are used to develop two separate IDSs, namely the Bayesian-based and the RPL specification-based, to deal with the two aforementioned threat types. We also evaluate the accuracy and energy-efficiency of our IDSs to ensure they can operate with little effect on 6LoWPAN performance.

1.4. Contribution of the Thesis

During the research, several contributions to knowledge have been emerged as follows.

Analyse and quantify the internal threats: we have analysed thoroughly 6LoWPAN security from different aspects focusing on internal attacks. We categorised such attacks into two types: those who aim at manipulating network performance targeting traffic, channel, or neighbour nodes (performance-type), and those who aim at breaking optimal network topology (topology-type). Our approach put attacks with similar nature regarding behaviours and targets into the same group to serve as guideline for justifying the relevant countermeasures. The categorisation also suggests that these two types of attacks need separate detection systems to deal with, because once type focuses only on node

behaviour, which related mainly to performance statistics, while the other type targets protocol operations, which will need particular rules to check on protocol behaviours. Based on the categorisation, we implemented and studied the impacts of typical 6LoWPAN internal threats, including the *Black Hole/Grey Hole, Jamming*, *Delaying*, and *Hello Flood* in the performance-type, the *Sinkhole, Rank, Local Repair*, and *DIS* attack in the topology type. Among them, the *Rank attack, the Local Repair attack, and the DIS attack* are novel 6LoWPAN threats that were proposed by us [8-10]. The study of such attacks not only show how severe the attacks can be, but also reveal the important and sensitive metrics to consider for detecting anomaly. These results are closely related to the construction of the two IDSs proposed in this thesis.

Develop a Bayesian-based IDS to monitor the network for detecting the performance-type internal threats: We proposed a solution to detect the performance-type of internal threats, which targeting traffic, channel, and other neighbours' behaviours. We first introduced an effective monitoring architecture that involves modern technologies like cloud computing and multiple interfaces of sensor devices with Wi-Fi, 4G, 3G, or GSM, to expand the IDS capability. This architecture help to deal with the heavy workload and communication of our proposed Bayesian-based solution, which were never thought to be feasible in 6LoWPAN before. Our Bayesian-based IDS mainly aim at collecting statistical data of the node behaviours to extract the essential features to feed in the Bayesian statistic model to calculate the probability that threats may happen. The main advantage of this method is the anomaly judgement based on a wider view through considering multiple monitoring metrics at the same time. We show that this method is effective in terms of detection ability, while it is also lightweight to be applied in the network.

Profile the RPL and develop a specification-based IDS based on this profile for detecting the topology-type internal threats: We proposed **a** solution to detect the topology-type of 6LoWPAN internal threats. RPL is the underlying routing protocol for 6LoWPAN, so building a specification-based IDS for RPL is one of the most efficient ways to detect quickly and accurately any 6LoWPAN attack that breaks its optimised topology. We propose a practical approach to semi-auto profile the RPL behaviours, which can also be applied to other protocol profiling. Based on the knowledge gained from this process, we built a specification-based IDS to secure the RPL and 6LoWPAN

from the internal topology attack. Our module showed high effective of detection rate on operation, while still saving resources due to the architecture that we apply. Our profiling method is also semi-auto, which add more flexible in building and constructing the protocol specifications.

1.5. Thesis Structure

The structure of the thesis is as follows:

In **Chapter 2**, we present an extensive literature review of the previous work in securing 6LoWPAN. The review first gives a background of the important concepts before discussing in depth the vulnerabilities of 6LoWPAN as well as potential countermeasures.

Chapter 3 describes the framework to assess the internal threats before giving details on our categorisation of aforementioned two types of internal threats. In each type, selected attacks are implemented while their impacts to network operation and attackers' behaviours are assessed. The results are analysed to obtain the essential knowledge of the attacks and justify sensitive metrics that can be monitored later to detect them.

Chapter 4 presents the process of building a Bayesian-based IDS to detect the performance-type internal threats. The general step-by-step procedure in constructing a Bayesian model for judging node behaviours is introduced. Different metrics are studied to form the Bayesian structure while simulation data is used to train and test the effectiveness of the Bayesian model. An efficient monitoring architecture is also proposed to extend the capability of the 6LoWPAN IDS.

Chapter 5 introduces a practical approach to profile the RPL operation into legitimate states, transitions, and corresponding statistic. Based on the profiled model, an RPL specification-based IDS is built to detect the 6LoWPAN topology-type internal threats.

Finally, **Chapter 6** draws together the results and conclusions of the whole thesis. Possible direction of the future research will also be discussed in this chapter.

CHAPTER 2. LITERATURE REVIEW

2.1. Introduction

6LoWPAN and its underlying routing protocol RPL are standards that have been recently developed from scratch to serve specific requirements in IoT. There have not been many internal threats constructed particularly to the standards yet. However, most of the internal threats in wireless sensor network (WSN) are feasible in 6LoWPAN due to their similarity in network nature. Consequently, operators can employ common WSN defence techniques to secure the network system.

In this chapter, we first introduce 6LoWPAN operation and its main routing protocol – RPL. Such knowledge is needed to understand the attack behaviours and design the defence system. We then discuss the standard's vulnerabilities; particularly focusing on the internal threats, and the prominent defending techniques. Cryptography and Intrusion Detection System (IDS) are commonly seen as the two lines of 6LoWPAN security. The former aims at protecting the system from external attackers, while the latter deals with monitoring the network to detect the internal attackers. As this thesis aim at detecting the internal threats, we will focus on the IDS solutions. In detail, different issues regarding IDS in 6LoWPAN will be considered. This chapter will form the fundamental basis for our research through pointing out the potential threats, choosing the suitable monitoring techniques, designing a suitable IDS architecture, and providing a framework to evaluate the solutions.

2.2. Overview of 6LoWPAN

6LoWPAN has been introduced recently, but it has attracted lot of interest from both academia and industry. The first two-6LoWPAN specifications, RFC 4919 and RFC 4944, were released in 2007. The former specifies 6LoWPAN requirements and goals, while the latter presents its format and functionalities. Other mechanisms of the standard have been improved like header compression, Neighbour Discovery, use cases and routing requirements. Zigbee Alliance, a research group specialising in the ad hoc and 802.15.4 network, announced that it would integrate IETF standards such as 6LoWPAN and RPL into its future specifications [11].

2.2.1. 6LoWPAN Topology

A 6LoWPAN network consists of one or more local LoWPANs, which are all connected by IPv6 to the Internet through a gateway (border router). The LoWPAN devices are characterised by short radio range, low data rate, low power and low cost, which requires optimising operation to save resource for maintaining node life. LoWPAN supports both star and peer-to-peer topologies; however, the topology can be changed frequently due to uncertain radio frequency, mobility, and battery drain.

Figure 2.1 shows that for IP network, IP is the only protocol used to connect data link and physical layer to upper layer. 6LoWPAN, on the other hand, utilises the 6LoWPAN stack, a combination of LoWPAN adaptation layer and IPv6, to connect its WSNs to the Internet. The biggest challenging aspect of this combination is to adapt the packet sizes between the two layers, which are 1280 octets in IPv6 and 127 octets in LoWPAN. 6LoWPAN implements the adaptation layer in the border router to process the adaptation by fragmenting the packets at IPv6 layer before reassembling them in WSN (802.15.4) layer. Besides, the Data Link and Physical layer use protocols specified for sensor devices while the Transport layer does not commonly use TCP due to performance efficiency and complexity [11].

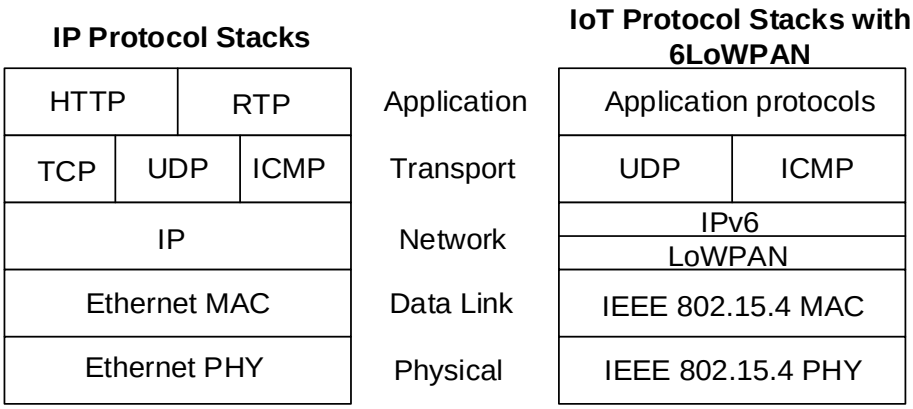


Figure 2.1. Comparison of 6LoWPAN and typical IP protocol stacks

In 2008, another IETF working group, Routing Over Low-power and Lossy network (ROLL), was formed to establish a solution for 6LoWPAN network layer. This group proposed RPL – Routing protocol for Low-power and Lossy network, which is later considered the underlying routing protocol for 6LoWPAN.

2.2.2. The RPL Framework

2.2.2.1. RPL Overview

Because 6LoWPAN is specifically designed, it also has special routing requirements, which are defined in RFC 5867 [12], 5826 [13], 5673 [14] and 5548 [15]. Routing protocol for 6LoWPAN must satisfy the following requirements [12-17]:

- Support different types of communication Unicast/anycast/multicast
- Adaptive routing with different network condition
- Constraint-based
- Support different traffic: multipoint-to-point (sensor nodes to sink manner), point-to-multipoint (sink broadcasts); and point-to-point traffic (sensor nodes communicate to each other)
- Scalability
- Configuration and management
- Node attribute
- Performance
- Security.

The ROLL working group extensively evaluated existing routing protocols, such as OSPF, OLSR, RIP, AODV, DSDV, DYMO, DSR, etc. and concluded that none of them can satisfy all requirements [18]. Therefore, ROLL proposed RPL, which was specified according to all these requirements. This protocol was then considered an underlying routing protocol for this network.

2.2.2.2. RPL Architecture and Operations

RPL components include sensor nodes, which act as hosts or intermediate routers for transmitting packets in WSN; and Local Border Router (LBR), which stays in the network edge and communicates through a common backbone such as a transit link [18], to translate packets through WSN to the Internet. 6LoWPAN nodes connect with a Directed Acyclic Graph (DAG) topology, which contains no loop. The DAG is then separated into multiple Destination Oriented Directed Acyclic Graphs (DODAGs). The roots of these DODAGs are normally LBR, which connected together and to the Internet through the backbone. Each DODAG is considered a logical configuration of physical node, so a node

can join multiple DODAGs to support routing optimisation [16]. Figure 2.2 illustrates this general architecture.

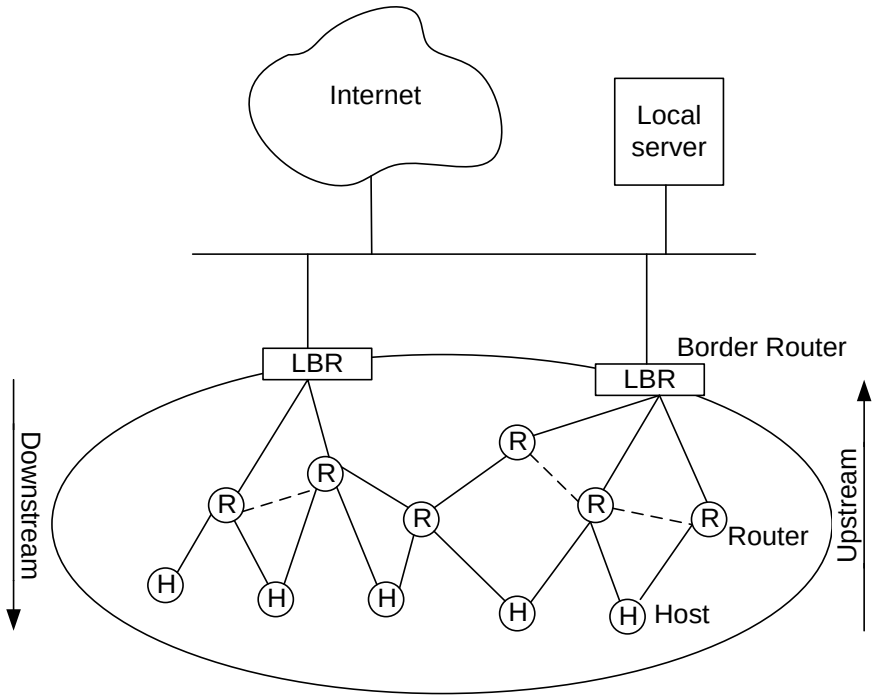


Figure 2.2. 6LoWPAN architecture

Nodes in DODAG select and optimise the routing path using some node/link metrics, called DODAG instances. Some examples of the metrics are: node state, node energy, hop count, throughput, latency, link reliability, link colour attribute [16]. Nodes inside each DODAG uses a specific metric, which is set in the Objective Code Point (OCP). They also share an Objective Function (OF), a function to calculate the value of the route towards the sink according to the selected OCP, to rank and select the route. Such value is represented through the *Rank* concept [18]. Nodes with the same rank can be sibling nodes, while consecutive rank nodes can be parents and child. Messages transmitting in RPL need to follow the *Rank rule*, which states that packets can only be transmitted by either upstream with node along the path having ranks strictly decrease, or downstream with ranks strictly increase. This rule was created to prevent routing loops.

In the establishment phase, the DODAG root starts broadcasting its DODAG Information Objective (DIO) messages, which contain information about its rank, OCP and DAG-ID. All root neighbours have a direct path toward the root, so they set their rank to 1, add the root's address as their parent's address, and broadcast this information in their own DIOs. Once other network nodes receive these DIOs, they form a set of parent nodes and select a preferred parent, which has the best rank among the set. This preferred parent will be

their default next hop to forward packet to the root. The nodes on their turns continue to calculate their own rank based on the rank of their parent and path cost, and form their own DIO, which includes rank, OCP and DODAG-ID, to broadcast. By repeating this broadcasting mechanism, DIOs are propagated throughout the network. Every node then has a preferred parent to transmit packet, and the DODAG topology is created.

When a node joins the network for the first time, it can either wait for a DIO or send a DODAG Information Solicitation message (DIS) to ask others sending DIO (if the waiting time is long). Once this node receives a DIO, it chooses its preferred parent and builds a Destination Advertisement Object (DAO) message, which contains its address and the parent's address as a prefix. The DAO is advertised for other nodes to update their routing tables or optimising their parents if possible.

RPL provides two mechanisms to fix the broken links in the maintenance phase. The first one, Global Repair, is started by DODAG root sending new DAG sequence number to reform the whole topology. Once nodes receive new DIO messages, they start parent selection and update link cost again. If a local node suffers from broken links and it does not want to wait a long time for Global Repair, it can use Local Repair mechanism. To do this, it needs to broadcast the poison message to its children informing that they need to find a new preferred parent. It then sends a DIS message to request the new topology information, and repeat everything like the first time it joins the network.

RPL uses the trickle algorithm for scheduling the DIOs broadcast to save resources. In this algorithm, each node maintains a trickle time and a DIO counter that serves as indicators for the stability of the topology. The "trickle time" interval will decide the moment when the node has to send its next DIOs. Each time a node receives a DIO without a change compared to the previous DIO; its DIO counter will be increased. Later, if the DIO counter exceeds a pre-set value called the "redundancy threshold", the node will reset its DIO counter and double the trickle time. The reason for increasing the trickle time is that the DIO counter threshold ensures the stability of the topology over an acceptable period, so there is less need of making frequent topology updates. This mechanism helps to reduce the number of DIOs generated in order to save network resources. On the other hand, if there is any change in the incoming DIO, the node will reset its DIO counter to zero and minimise its trigger time. This will allow the network to update its topology quickly through fast DIO generation.

2.2.3. 6LoWPAN Security Requirements

RFC4919 [19] specifies a list of security requirements for 6LoWPAN, which mainly aim at protecting the communications from the end-users to the sensor network. The detailed requirements are:

- Confidentiality: only authorised users can access the information
- Authentication: data have only originated from a trusted source
- Integrity: the received data remain unchanged during transmission
- Freshness: considers for both data and key to ensure no replayed of old messages
- Availability: guarantees that the data can be accessible when needed
- Robustness: providing operation despite the abnormal conditions
- Resiliency: provides an acceptable level of security even in the case which some nodes are compromised
- Energy efficiency: reduces the control overhead to maximise network lifetime
- Assurance: the ability to disseminate different information

These requirements require the combination of different securing approaches. Cryptography is considered the first line defense that protecting confidentiality, authentication and integrity. This line, however, cannot guarantee other QoS requirements like availability, robustness, and resiliency. Therefore, Intrusion Detection System (IDS) needs to be used as the second line to monitor and detect the malicious sources from the early phase to eliminate long-term damage from the attacks.

2.3. Vulnerabilities in 6LoWPAN Security

6LoWPAN is the combination of IPv6 and WSN, so security threats from both sides need to be examined. There are also threats towards the adaptation layer to attack the packet translation process. On the other hand, the operation of 6LoWPAN is affected by the RPL performance, so analysis of threats towards this protocol is also essential.

2.3.1. Security Threats from the Internet Side

End-users from the Internet can access information from the sensor once 6LoWPAN is implemented. This raises authentication threats, the availability of sensor network, and user accountability. The adversaries can access the information illegally if no

authentication mechanism is applied. When a communication channel between end-user and sensor network is established, the attackers can also eavesdrop the sensitive information from the data stream. Besides, the users accountability to access sensor network should also be considered [20]. The availability of the communication should be guaranteed by protecting the sensor side and adapting the operation of Internet side which has resource constraint nature.

Another type of threat happens when the attackers can get control of the sensor nodes through the Internet. For example, the botnet attack [21] creates a botnet inside the sensor network for forging the data sending to the sink. The botnet falsifies the data in the user-end that leads to wrong alarm or decision. Although the sensor botnet does not have enough resources for making a successful DDoS attack to other networks; attackers can make a DDoS attack to the botnet itself by flooding to drain the resource.

Cryptography alone cannot defend the DoS attack from the Internet to the sensor network, so there is a need for implementing an IDS for analysing the IP traffic between the two. Traditional IDS solutions in the Internet or in the sensor network cannot be simply applied because of the dissimilarity of traffic pattern.

2.3.2. Security Threats from the Adaptation Layer

The adaptation layer is implemented at the border router for translating the packet between the two networks. The border router is normally a computer with wired connection to the Internet and has strong security protection. However, its packet fragmentation and reassembly process still have some vulnerability.

Kim [7] proposed that fragmentation attack from the IP network can be applied to this layer by modifying or reconstructing the packet fragmentation fields like datagram size, datagram tag or datagram offset. Examples of the threats are Tiny Fragmentation, Ping of Death, Jolt, Teardrop, bank, New Teardrop, or Frag router attack [7]. These attacks can cause critical damage to sensor nodes, for instance, re-assemble buffer overflow due to packet re-sequence; exhaust the resource because of processing unnecessary fragmentation; or shut down, reboot the sensor nodes.

2.3.3. Security Threats from WSN Side

The security threats of WSN have been extensively studied by the research community. The attackers can be classified by several schemes: internal – external, passive – active, compromising methods, host-based or network-based [22, 23]. The most common approach to differentiate an attacker is by identifying whether it initiates from internal or external source. An internal attacker uses a source that can participate fully in the network communication. This source is either a legitimate node being compromised or a device, which gains access by having network secret key. The source can stay either in WSN or in the Internet side. On the other hand, external attackers can only have access to network from outside through listening or eavesdropping, so its manipulation to the network operation is limited compared with the internal attackers. Consequently, defending the WSN side from the internal attackers is much more demanding than from the external. The other ways to identify an attacker are through their attack approach, which means it can be either passive or active, host-based or network-based. For example, a passive attacker is the one that mainly manipulates the system based on observing the communication instead of interacting with other devices. Another example of identification, an active attacker tries to affect to other nodes' operation without concerns about its activities being spotted.

Techniques for protecting the network from the internal and external attackers are different. The external attackers usually use unauthorised listening or Denial-of-Services attacks, therefore, the main protecting technique is cryptography. On the other hand, insider malicious nodes can be created by several ways: attackers physically capture the nodes and reprogram them, attackers use software and devices to breach the cryptographic key or inject malicious code [24]. In such cases, the attackers have all the keys, so they can easily overcome any cryptography test. However, the consequences of the internal attacks are usually the downgrading of network performance, so the IDS approach is more suitable to detect the anomaly performance in the early stage.

A common way to categorise different types of attack is to group them into the targeting layer. A summary of one layer-approach categorisation, which were obtained from [22, 24, 25], is given in the Figure 2.3. Some of these threats are more dangerous as they can be deployed easily and lead to sophisticated attacks. If the system cannot identify them early, their effects on network operation may add up in the long-term. One example is the

Sybil attack, which uses the packet forging mechanism. Undetected Sybil nodes can initiate other attacks like misdirection, exhaustion, unfairness [22], and creates a more severe downgrade of network performance. For example, those attacks can make WSN unavailable, partitioned, or resource exhausted. Another dangerous attack is the Sinkhole, which uses a packet dropping mechanism to attract traffic to specific node. If the system cannot detect early, it can generate selective forwarding, black hole attack and combines to partition the network [22].

Attackers can also apply techniques to attack some IPv6 mechanisms like Neighbour Discovery and Address Auto-configuration in WSN [26]. If the attackers can bypass these mechanisms to spoof the neighbour solicitation/advertisement or the redirect messages, they may degrade the routing performance by falsifying the members' views on topology.

From Figure 2.3, it can be seen that most of the WSN threats focus on the network layer and aims at degrading the network operation. Therefore, it is necessary to assess further the routing security, which is RPL in more detail.

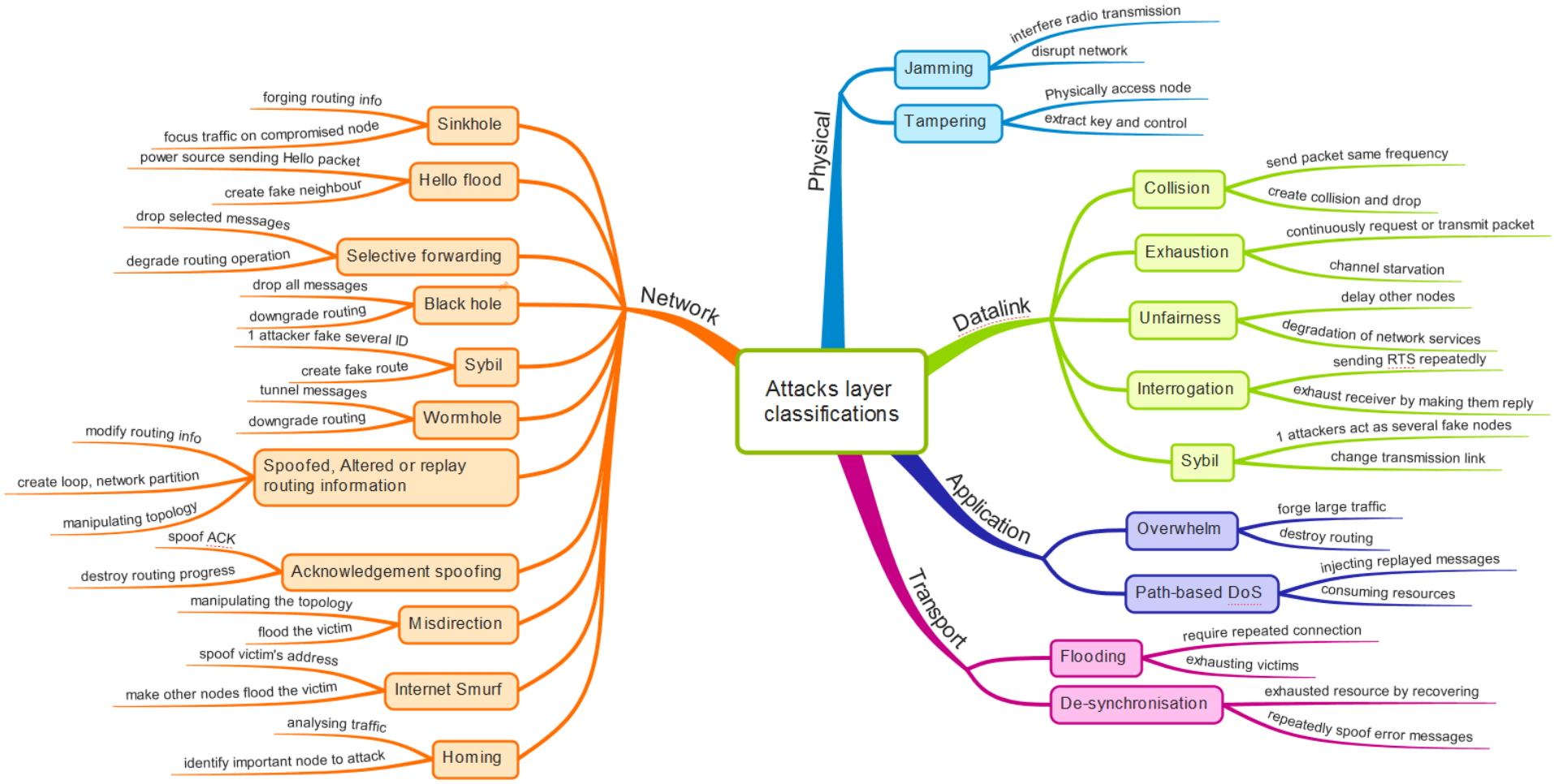


Figure 2.3. Illustration of potential WSN attacks in 6LoWPAN

2.3.4. Security Threats from the RPL

The drawbacks of 6LoWAPN security, such as weak communication links and non tamper-resistant nodes, make RPL weak from internal attack. Once a benign node becomes an internal adversary, it can break the network operation without being detected by cryptographic mechanisms.

In [27], the authors provided a collection of attacks towards RPL. RPL attacks are divided into three groups according to their target, namely Resources, Topology, and Traffic, as can be seen in Figure 2.4. According to the authors, attacks towards RPL resource are those who aim at making legitimate nodes to perform unnecessary processing in order to exhaust their resources; while topology attackers are those who target either making RPL sub-optimal or isolated; or eavesdropping and misleading traffic. Such RPL attacks taxonomy may show some overlaps. For example, a flooding attack may belong to both resource and traffic group, as it exhausts the nodes' resource and creates misappropriations at the same time. Besides, the Rank attack, which manipulating the Rank property, can deplete network resources, break optimal topology, and generate bad traffic altogether.

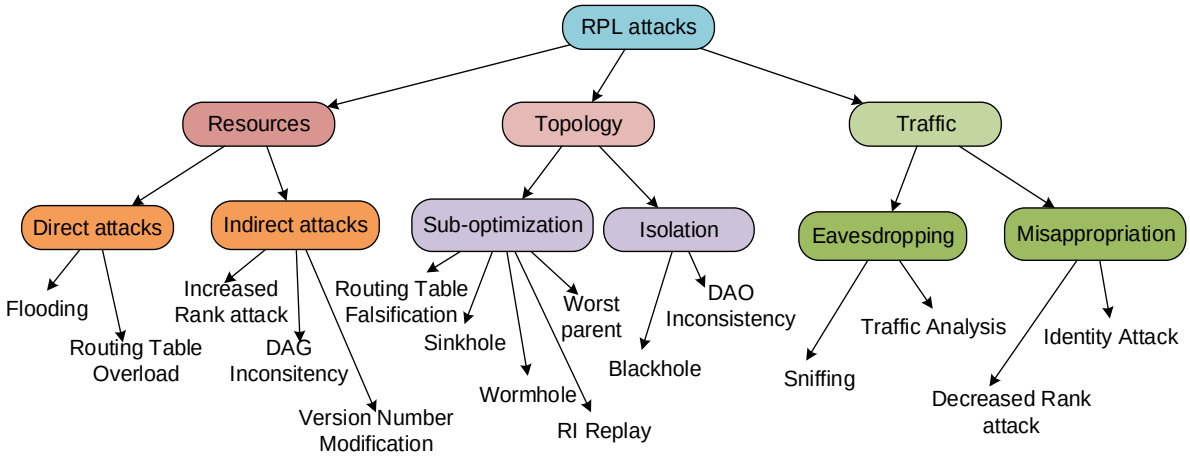


Figure 2.4. Taxonomy of attacks against RPL [27]

Our view on RPL attacks is different. We consider RPL a routing protocol, so its main objective is to establish and maintain an optimal network topology. Other attacks not targeting the optimal network topology will not directly relate to RPL operation. Based on that view, we categorised the RPL attacks according to the specific RPL properties that the attackers target as can be seen in Figure 2.5.

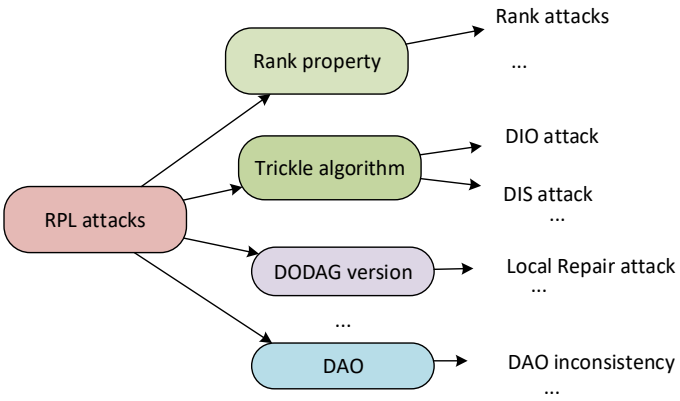


Figure 2.5. Taxonomy of RPL attacks based on specific property targets

Our research of RPL attacks is one of the first works that analysed and quantified specific threats towards RPL performance. In [8-10], we proposed different kinds of RPL attacks, including Rank, Local Repair, DIS, DAO, neighbour, and DIO attacks. Such attacks aim at specific mechanisms of RPL such as Rank property, trickle procedure, local repair and so on. Similar research from other authors are the work in [28, 29], which focus on the Rank property; and [30] emphasises on DODAG version (similar to local repair attack). Such work will be discussed further in Chapter 3.

2.3.5. Categorisation of Internal Attackers

We categorise the internal attacks into two major types: the performance and the topology attacks. The performance attacks are those that target downgrading network performance directly. For example, they can decrease the delivery ratio as in Black Hole or Selective Forwarding attacks, or increase the delay as in the delay attack. Considering the objectives of such threats, this type can be further divided into three other sub-categories, which aim at (i) traffic through a node (Black Hole, Grey Hole, Selective Forwarding, …); (ii) channel (Jamming, Collision, Exhaustion, …); or (iii) interfering the neighbours (Hello Flood, Sybil, …). On the other hand, topology attacks are those that aim at disrupting the RPL optimal topology, which later indirectly affect network performance. In our view, any internal attacks will fall into one of these two categories. If a network is free from these two types, it will always operate with the optimal topology while every node performs optimally, then this network can be considered in ideal working condition.

Some typical attacks in these two categories will be quantified and analysed in detail in Chapter 3. Our solutions will also aim at detecting such attacks by using the knowledge obtained from analysing their behaviours. We believe the ultimate aims of monitoring a

network are to ensure that this network always has an optimal topology, while all nodes working with optimal performance.

The next sections will review the two common defence approaches that can be applied, which are the cryptography as the first line to prevent external attackers, and the Intrusion Detection System (IDS) to detect the internal attacker sources.

2.4. Cryptography Techniques

Cryptography always involves encrypting the messages in operation. The approach aims at protecting the following points:

- Authentication: only authenticated user, who has the right key, can decrypt and read the messages
- Integrity: message content should not be changed during transmission
- Confidentiality: no one can understand the message without the key.

We will consider these cryptographic requirements in both the WSN and IPv6 side of 6LoWPAN.

Common cryptographic approaches in WSN use Advance Encryption Standard (AES) for securing the link layer with several operating modes. However, most of the modes do not ensure integrity requirements [19]. Message authentication and encryption mechanisms could be implemented by the transceiver chip of specialised hardware. Nevertheless, given the resource constraint of sensor nodes, the processing time and overhead created by such mechanisms may downgrade network performance or shorten nodes' lifetime. Hence, a lot of research work have focused on finding more lightweight cryptographic solutions for WSN recently, such as, TinyEEC [31] and NanoEEC [32]. These improvements may help to provide viable encryptions for link layer messages; however, the link-layer mechanism only ensures hop-by-hop security, while the network requires secured end-to-end communication. Moreover, there is no authentication or key management to support the host. Several key distribution methods like pre-distribute, key pool, have been proposed for WSN, but they lack of scalablity. Another vulnerability of the key management system is that at the bootstrap time, an adversary can sit among other nodes without being required to authenticate, so they can obtain the keys freely.

From the IPv6 side, common cryptographic solutions use IPsec to secure data exchange. It looks promising to extend IPsec to 6LoWPAN because if it can be done, an end-to-end

security will be established from the Internet to WSN. In addition, there will be no need for a trustworthy gateway. However, the public key cryptography encryption in IPSec is too heavy for resource-constrained devices in WSN. Besides, exchang key is also a challenge. The Internet Key Exchange from IPsec is not feasible because of using heavy signalling messages, which do not fit the small WSN packet size and the energy efficiency requirement.

Recent research has made significant efforts in transforming IPSec into a feasible solution for 6LoWPAN. For example, authors in [33, 34] showed ways to combine RSA (Rivest-Shamir-Adleman) and ECC (Elliptical Curve Cryptography) techniques for a light weight and adaptable encryption. Liu and Ning proposed pairwise key pre-distribution [35] and DHB-KEY [36] for simplifying the key distributions. Raza *et al.* [37] provided a 6LoWPAN-IPsec Specification including definitions for Authentication Header (AH) to safeguard the integrity of the whole IPv6 datagram, including application data, IPv6 headers and Encapsulation Security Payload (ESP) extension headers for securing application data. With such improvements, IPsec was thought to be a potential security solution for IP based WSN [38-40].

Although research shows significant improvements in using cryptography in 6LoWPAN, its application still has to overcome many issues. Moreover, the approach is only helpful while protecting 6LoWPAN from external attacks, but lack ability to detect and eliminate the internal attacks. This is because cryptography cannot detect attackers who can show the legal keys. A network using only cryptography is therefore, weak against internal attacks aiming at network performance such as DoS or battery, resource attacks like jamming, exhaust attack.

Overall, cryptography alone cannot provide total security for 6LoWPAN. There is a need for implementing an IDS in order to monitor internal attackers to prevent long-term damage effect. IDS is an efficient way for detecting attacker bypassing the cryptography line, as well as monitoring and ensuring a normal network performance.

2.5. IDS Techniques

IDS aims to detect internal attacks by analysing and identifying their abnormal behaviours. Once the anomaly is detected, the IDS can raise alarm and deploy appropriate mechanisms to eliminate the attack sources. Since the internal attackers are assumed to have all the keys to operate legitimately as the normal nodes, the cryptography line

defence will not be able to detect them. As a result, 6LoWPAN will need IDS as a second defence line to detect any attackers that passed the first line.

2.5.1. Overview of IDS

IDS is a well-known network security approach which has attracted a lot of research interest since the 70s decade [41]. The technology development has changed the communication environment from wired, wireless, Ad hoc to sensor network recently. IDS solutions have also changed from feature selection, data collection, and analysis techniques to adapt to network environment. WSN nature is different from other networks in terms of device characteristics and resource availability. Hence, an IDS working in WSN should also be optimised regarding these features, data collection and computational work. Moreover, IDS in 6LoWPAN need to be more optimised due to the requirements of network scalability.

IDS approaches are often divided by misuse, anomaly-based and specification-based type. A misuse IDS defines patterns of the known attacks. When monitoring the network operation, if it discovers any data that matches the pattern it will raise alarm. This method can provide low false-alarm rate, but it is not favoured because 6LoWPAN internal attacks are not be well defined yet.

Another method, anomaly-based IDS focuses on classifying the normal network behaviours, then monitor and compare to detect any anomaly. The method computes the deviations between the monitored data and the legitimate pattern. If the deviation exceeds a threshold, it will raise an alarm. Anomaly-based IDS has the ability to detect new attacks because it considers the performance deviation rather than specific behaviours. It also does not consume many resources. However, the false-alarm rate is still high because the system cannot differentiate clearly between the misuse and malicious behaviours.

On the other hand, the specification-based IDS profiles the normal operations of the network in detail and monitors any behaviours that break this specification. The profiling works are usually done by specialists. Specification-based IDS also has the ability to detect the new attacks, if these attacks make the network behave differently from the patterns. The accuracy of the specification-based tends to be higher than the anomaly-based because the comparison is concrete and clear. However, its disadvantages lie on the needs of expertise, which will not be flexible when the system needs to upgraded due to discovering new vulnerability.

In the next section, we will review different IDS solutions for WSN, categorising into two main groups: anomaly and specification-based.

2.5.2. Anomaly-based IDS

This section reviews some of the most popular anomaly-based techniques for WSN, which are Artificial Intelligent, Data Mining, Agent Based IDS, and Statistics-based approach.

2.5.2.1. Artificial Intelligent

Artificial Intelligent (AI) techniques have been applied widely in wired IDS and now the solutions are moving to the WSN. The main application techniques are (i) Semantic-based (ii) Fuzzy Logic (iii) Game Theory, and (iv) Bio-inspired.

In the semantic-based technique, the solutions focus on extracting the features of WSN and constructing security ontology to build formal semantics for the network. The semantics are then normalized for checking performance. Examples in this direction are the work of Mao [42] and Chen [43]. Mao [42] defined four layers of the network, which are network, semantic, model and co-operative layer, and presented the relations between these layers as a pattern to check the normal operation. On the other hand, Chen [43] transferred WSN nodes into the ontology concept and calculated the relationship of the whole network to define threshold in relationship. This threshold serves as a reference for the node to monitor and discover any of its anomaly neighbours. This approach is, however, difficult to build in a diverse 6LoWPAN environment, where the system has a wide range of node types and relationships.

Choosing a threshold value for the behaviours is also an important issue in IDS. If the threshold is too low, the detection sensitivity may be low because of recognising normal nodes as attackers. Moreover, if it is too high, the system accuracy may not high because of not being able to detect the threats. For these reasons, the fuzzy logic approach is used for setting a dynamic IDS threshold.

Lee et al. [44] used the number of cluster nodes, the value of the key dissemination limit, and the distance from the base station to each cluster to calculate the fuzzy threshold and broadcast it periodically. This method adapts the topology changes due to the movement of the nodes so it can drop false reports. However, it requires the BS to store and calculate

the distance toward cluster nodes and energy consumption so it still consumes many resources.

Sang and Tae [45] used four factors: the node energy level, neighbour node list, message transmission rate, and error rate in the transmission to calculate the dynamic threshold for detecting DoS attacks. The integrated fuzzy threshold is easy to be calculated. However, the threshold for each parameter is chosen by experiment, so there is no guarantee that the solution will work in a different network environment.

Parekh and Cam [46] used a Directed Acyclic Graph and Probability table to represent the dynamic site condition to calculate the threshold value for minimizing the false alarm rate. The sensors are selectively chosen to assign weights to their sensed reading so that they can improve the quality of detection. The disadvantage of this method is that it requires knowing the network topology as well as the roles of sensor nodes, so it will decrease the scalability when implementing in 6LoWPAN.

The fuzzy approach provides an adaptive way to deal with the environment changes and improves the accuracy of the network, but it needs a stronger theoretical model for dealing with different network environments in 6LoWPAN.

Game theory solutions, on the other hand, aim at modelling the network security as a game between players with contradiction objectives. The game type can be either non-cooperative or cooperative, zero-sum or non-zero-sum. The objective is to discover the optimised strategies for the players, called the Nash Equilibrium.

Rong et al. [47] defined a simple payoff matrix with probability measures for the IDS to protect important nodes in the network effectively from the DoS attack. Estiri [48] proposed a repeated game model for detecting the dropping packet attacks which reward the node reputation every time it forwards and cooperates, while punishing every time it does not. After a number of repeated times, the average number of packets dropped is shown to get to a stable level and the malicious nodes either to stop the attack or to be exploited. Estiri [49] also proposed a Bayesian game with incomplete information to present the interaction between malicious and normal nodes in terms of signalling.

The gaming approach is a strong tool and promising for improving the detection accuracy. However, some main concerns of this approach are the rational assumption of the players;

the complexity of modelling the real network; and the large computation that consumes significant WSN resources.

The Bio-inspired approaches migrate from the animal behaviours and model these for optimising the security solutions. Banerjee et al. [50] combines the Emotional Ants and the conventional machine learning technique for keeping track of the intruder trials. The IDS agent works as the ant agent and later is transformed to be the emotional ant agent for making decisions. The main advantages of this solution are the ability to perceive behavioural patterns, deliberate and act based on a self-organising principle combining with probability values.

Soroush [51] also used a Boosting Ant Colony based Data Mining for extracting a classification rule set from a network dataset. The pheromone and Entropy function are used to direct each tour of the ants and iteratively continues to extract a final set of rules, which were later used as detection patterns in the larger dataset. This method is an effective way to mine the data; however, it consumes time and resource to obtain the results.

Overall, the main advantage of the AI techniques is the ability to extract value information about malicious from the data with high accuracy. Its main drawback is, nevertheless, to consume a large value of resource on training and testing data.

2.5.2.2. Data Mining

The Data Mining approach mainly apply machine-learning techniques to derive the detection rules. In this approach, the system is implemented with a distributed configuration. In order to reach a high accuracy, it requires great computational power and a large memory space. Some techniques in this direction focus on classifying the data in order to reduce the IDS analysis work.

Xiong et al. [52] proposed the Support Vector Machine (SVM) technique to classify the feature subset as a positive feedback factors. Such factors can be adjusted for later use in Ant Colony Optimisation. The method reduced the feature subset while improving the classification accuracy.

Kaplantis et al. [53] also used SVM with polynomial kernel or Radial Basis Function (RBF) model for detecting Selective Forwarding and Black Hole attacks. The chosen parameters for monitoring are bandwidth and hop count within a sliding window. This

solution minimised the false positive rate, however, it consumes singificant resources in computing and communicating, which prevents the scalability and adaptability to network environment.

2.5.2.3. Agent-based IDS

Agent-based IDS provide a technique to split the workload through distributed IDS so that it can accelerate IDS operation. There are two common types of agent-based methods: autonomous distributed agents and mobile agents.

In the autonomous distributed agent-based IDS, agents can keep track of traffic while sharing information with other agents [54]. Zhang et al. [55] reports the implementing of a multi-agent based IDS, where they considered four types of agents: Basic agent, Coordination agent, Global Coordination agent, and Interface agents. Each agent runs a different task and has its own sub-categories. For instance, the basic agent includes Workstation agents, Network segment agents and Public server agents, which work on the workstations, the subnet level, and public server level respectively. In this way, any complex system will be divided into much simpler systems to manage more simply. However, such uses of many agents also increases the overhead, computational bottleneck as well as transmission delay generating by the communication between the agents.

The other method is the mobile agents, which are similar to distributed agents, but they also can move throughout the network to detect attacks. A mobile agent is a self-controlling program segment moving from node to node, doing both data transmission and IDS computation. In mobile agent based computing paradigms, a task specific executable code traverses the relevant sources to collect the data. Mobile agents can reduce the communication cost greatly by relocating the processing function to the data, instead of bringing the data to a central processor [56]. Authors in [57] proposed the mobile agent approach to detect the Sinkhole attack in AODV protocol. Their mobile agents inform sensor nodes of their legitimate neighbours so they will not listen to the traffics generated by malicious ones. However, since the mobile agents travel the whole network only once, some malicious nodes cannot be detected by the mobile agents. Furthermore, the effectiveness of the detection system will rely on the travelling path of the mobile agents, which requires optimisation of this path.

2.5.2.4. Statistical-based IDS

The statistical methods using several mathematical models for analysing the dataset to identify a threshold pattern to detect the anomaly behaviours. Some of the main techniques are (i) Mathematical model, (ii) Hidden Markov chain, and (iii) Bayesian Network.

The mathematical approaches use some statistical models such as linear or non-linear. Phuong et al. [58] used the Cumulative Sum to detect changes based on the cumulative effect of the changes made in the random sequence instead of checking every variable threshold. The model is easy to compute, strong, light-weighted and not resource consuming. However, the model accuracy rate is not high due to the lack of cooperation between the monitored nodes.

Ponomarchuk [59] analysed the number of received packets in a time window of a given length and inter-arrival time of packets for detecting anomaly behaviours. The packet Reception Rate was calculated based on the binomial distribution, while the inter-arrival time was based on the exponential distribution. The method provides low computation cost and low memory requirements for storing data. However, the model does not take into account the effect of the wireless network environments, which is crucial in practical implementation of 6LoWPAN.

The Hidden Markov Model (HMM) technique can profile the normal and abnormal patterns when analysing the data. Song et al. [60] used a Weak HMM, which is a non-parametric version of HMM to state the transition probabilities to loosen the rules of reach ability. The detecting mechanism is done by the scoring scheme and the deviation alarm. This approach showed the effectiveness in detecting several kinds of attacks, but the false positive error rate is high and the system still requires a large amount of resources. This needs to be improved before applying in the 6LoWPAN environments.

A Bayesian network (BN) is a technique that reflects the interactions between variables which represent primary causes and their relations that lead to the final outcome for a query [61]. In general, BN can be considered as graphical probabilistic models for multivariate analysis. The model is formed as a directed acyclic graph, which have associated probability distribution functions. The variables are characterised by the graph nodes, while relations between these variables are illustrated by the graph edges. The strengths of such relations are described through the Conditional Probability Tables.

BN has been applied successfully and widely in different network security issues. There are several work applying BN in WSN. Momani [62] combines the data trust and communication trust to infer the overall trust between nodes. The author showed the need to combine these two trust values for preventing misleading or breaking down threats of the network. The new trust model is represented by the Bayesian Fusion Algorithm, which combines these two trust values. Building the trust value for each node is important to show its reputation. The reputed sources are then can be used to justify which node is malicious.

Another direction of applying Bayesian method is to clarify the relationship between the network operation parameters to the attack possibility. When the system has a reference model of this relationship, it can detect the attack sources through the collected data. An example of this approach is the work in [63], which used a Bayesian Trust Model for calculating the MAC sub-layer data of WSN to mitigate the unfairness and consequent created by the DoS attacks. This solution can be generalised and adapted to other protocols by adjusting the networks and trust model parameters.

BN has several advantages that make it suitable to be utilised in detecting the performance-type attacks. First, this model can be trained through the historical performance data [64]. In detail, we can set up and simulate a number of common network scenarios, obtain the trace data, and extract the relevant metrics to study the relations. Such relations between these metrics can also be assessed through statistic model. Second, the model can predict even when some event observations are missing. For example, at the time of decision, if the system is unable to observe the state of some events, it can still give brief predictions based on the remaining events. Third, the model can provide the internal relations between the metrics. This brings a more robust way to justify the events to compare with methods, which considering only a specific variable.

The core of this method is the “Bayes’ theorem” introduced by Reverend Thomas Bayes in 1763, which allowed to obtain or update the probability of a hypothesis to be true, given a number of observations. The Bayes’s theorem is presented as follows. Given A and B as two events to be observed. Assume that P(A), P(B) is the probability of the event A or B occur respectively, P(A, B) is the probability when both the event A and B occur, while P(A/B), P(B/A) is the event A occur if B occur, and the event B occur if A occur, respectively. Then we have:

$$P(A,B) = P(A) \ * \ P\left(\frac{B}{A}\right) = P(B) * \ P(A/B)$$

$$\text{Therefore } P(A/B) \ = \frac{P\left(\frac{B}{A}\right) * P(A)}{P(B)}$$

The probability results will change according to the number of observations (reflecting through the corresponding probability) revealed. This model can give brief result even if only a small number of evidence observed. On the other hand, the more evidences the system absorbs, the more accurate it will be.

We present a brief example of forming and predicting using the Bayesian model as a background for readers to be familiar with the method, as we will apply it for detecting the performance internal threats later. In Figure 2.6 (a), the Bayesian model is given with four events to observe, which are Cloudy, Rain, Sprinkle, and Wet Floor; and one event to predict, i.e. Slippery. The arrows represent the cause-effect relationships between such events, for example, the Cloudy may cause the Rain or Sprinkler. The details of such relationships are described through the CPT given besides each node. Figure 2.6 (b) gives the prediction of the Slippery event (83.1%) when there is no evidence of the four observed evidence. Figure 2.6(c) shows there is 86.42% of Slippery if the Cloudy is observed. However, if the Cloudy is observed (Cloudy = T) and Sprinkler and Rain are observed to not occur (Sprinkle = F and Rain = F), then the chance of Slippery will decrease to 27%, which is shown in Figure 2.6 (d).

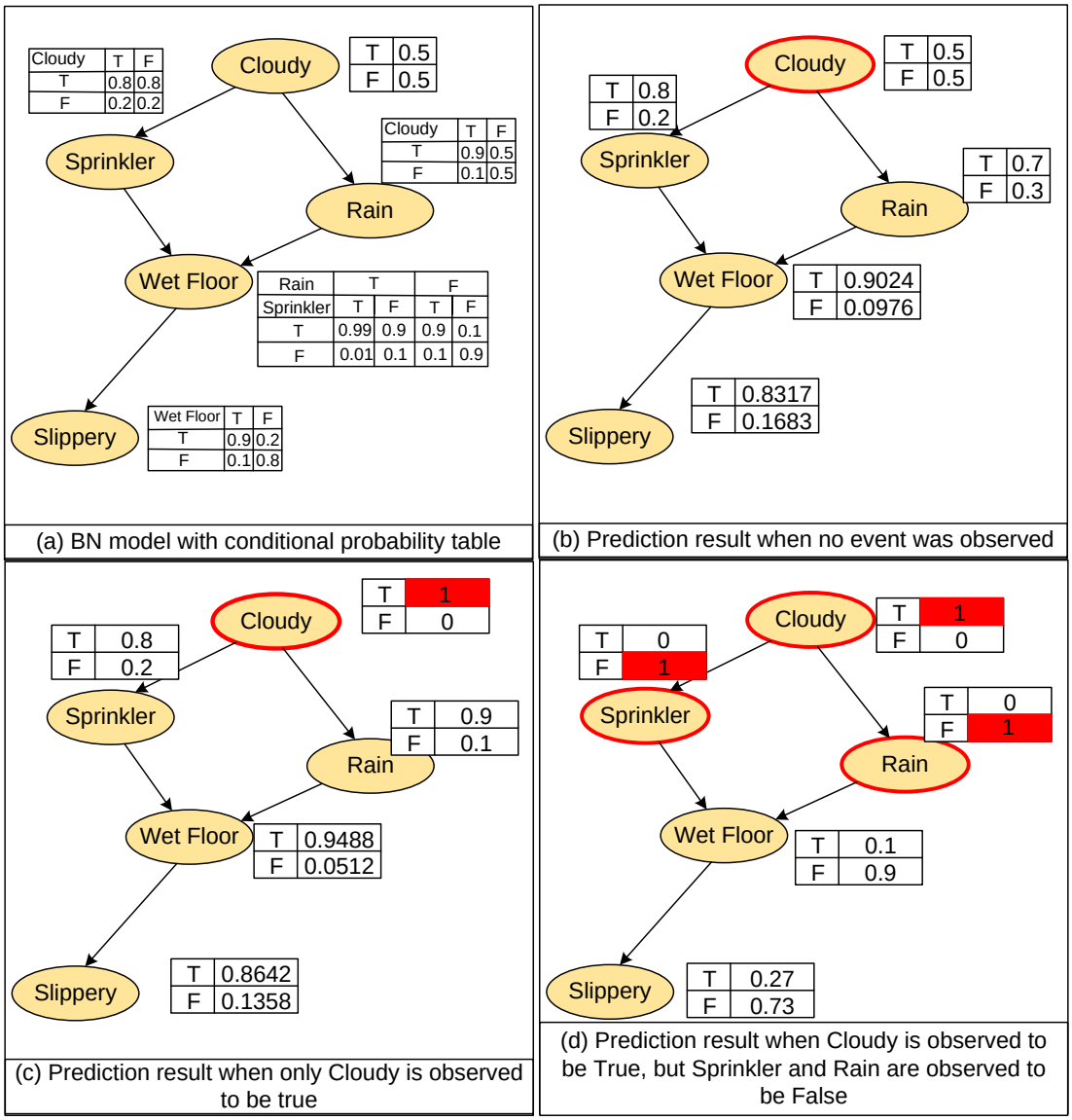


Figure 2.6. An example of the using Bayesian model to predict

Overall, anomaly IDS can adapt to different network environments, attacks and co-ordination attacks. Due to the difficulty in differentiating between misbehaviour and malicious nodes, it usually has a high false alarm rate. Other disadvantages include the time-consuming for analysing a large amount of data and the possibility that adversary can re-train the system to accept attack behaviours. Among a number of anomaly-based IDS we have reviewed, we prefer the Bayesian approach to detect the performance-type attacks in 6LoWPAN because of its outperforms in judging based on insight understanding of the system and the use of training data.

2.5.3. Specification-based Approach

Specification-based approach can derive the abstraction of the network operation; hence, it can simplify the feature selection and tailor monitoring to the needs of their own systems. Moreover, it can scale well and simplify the test operation for justifying a set of events, which constitutes a violation. It can also take advantage of the knowledge about possible attacks of system operators , and provide accurate attack detection with low false alarm rates [65, 66].

The main techniques used for these specifications are the state machine transitions, machine learning for pattern recognition and statistical analysis to derive automatically program specifications [65, 66]. In the literature, there are many specifications on some protocols working in environment similar to 6LoWPAN such as AODV, OLSR and CoP (Connectionless Routing Protocol).

Ning et al, [67] analysed changes on AODV operations in attack conditions. They mainly focused on the fields in the two messages RREQ and RRPL. Authors in [68] showed that some fields of the Route Request and Route Reply messages such as ID, Hop Count, header and Sequence Number can lead to threats like Man-in-the-middle or Tunnelling attack. Moreover, Grönkvist et al. [69] added other attacks like Forged Sequence number and Forged Hop count. Based on those analyses, the work in [69] provided different ways to specify the Route Request and Route Reply messages of the protocol based on the Finite State Machine (FSM) technique. The main idea is to analyse the received messages for detecting anomaly in transitions, which are defined in the threat identifications.

Tseng et al. [70] proposed an Optimized Link State Routing Protocol specification, which used an extended FSM technique with a backward checking algorithm to determine the corresponding transitions from the last event. Possible changes in the fields of Hello and topology control messages are also defined. The state transition analysis was also used for modelling host and network based intrusions. Orset et al. [71] proposed an extended FSM solution, which specify the formal specifications of the correct OLSR behaviours and uses a backward checking algorithm to detect run-time violations of the implementations. The authors develop several semantic rules for checking the specifications more quickly.

Mostarda [72] specified the operation of CoP by defining a Global Automaton based on some basic routing properties. The system can check the node states based on this

Automaton. The authors mentioned two ways to monitor the transitions, by either adding a field in the control messages to show the transition state, or sniffing to find the unique chain of rules that matches the sequence of invocations. Some semantic rules were also defined for simplifying the checking progress.

In order to apply specification-based IDS to 6LoWPAN, RPL operations need to be profiled. The prominent attacks towards RPL also need to be analysed to extract specific changes they will create for this specification model. If we can profile the operation of RPL accurately, the specification-based approach will be suitable because the detection issue will be simplified by verifying node behaviours with this profile.

A summary of different IDS for solutions is given in Figure 2.7 below.

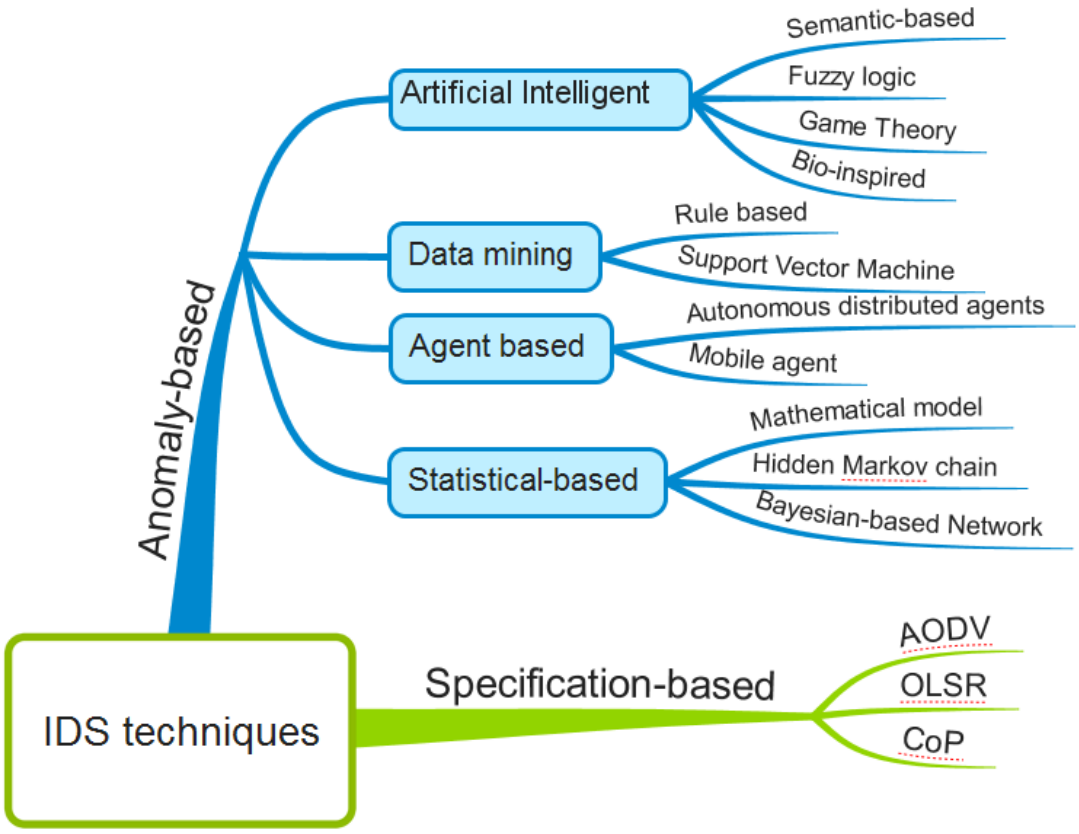


Figure 2.7. Summary of different IDS solutions for WSN

In the next section, we will discuss some main issues when building an IDS for 6LoWPAN.

2.6. Issues on Building IDS in 6LoWPAN

The previous sections shows that the most effective solution to detect 6LoWPAN internal threats is to monitor network behaviours. However, on building an IDS for 6LoWPAN, there are several issues that have not been well studied, including:

- The impacts of internal threats on RPL network and node performance have not been studied well enough
- The effective features for detecting the internal threats have not been justified well enough
- The legitimate behaviours of the RPL have not been profiled
- The design of IDS to match 6LoWPAN operation is not identified
- The approach to evaluate the effectiveness of the IDS is not clarified

These issues will be discussed in detail in this section.

2.6.1. Studying the Impact of the Internal Threats

Studying the impact of the internal threats on the network performance in general and node behaviour in particular, is important because such work help to understand the damaging level of the attacks. There is not much research of attack impacts on 6LoWPAN, despite many similar studies available widely in other similar networks. Due to the relative newness of 6LoWPAN/RPL design as well as the discovery of several novel attacks on such network, the impacts of the internal threats may not be the same like in other network.

Study of the attack impact can involve many factors. Weerasinghe and Fu [73] studied the impact of Black Hole attack on mobile Ad hoc network showed that the number of attackers, the size of the network (number of network nodes and the terrain set up), and node properties may affect attack impact significantly. Riecker et al [74] observed that topology setup and traffic are also among the factors that may change the impact. For example, in the Jamming attack, a sparse topology will show more severe impact on the percentage of dropped packet rather than a dense topology. In [9], we also show that even in a grid topology where most of the positions in the setup topology have the same node density and node property, Rank attack initiated in different positions of the network will create different impacts. As a result, it is complicated to study all the potential impact of an attack if considering all the relevant potential influencing factors.

The main reason of the diffrences on studying the attack impact on different network setup lies in the way we consider the impact. Assessment of impact on a particular metric is done through measuring the deviation of this metric between the normal and attacked scenario. In some of the network conditions, the benign node may behave similarly to the malicious node. For instance, in a dense network, the benign nodes tend to have more traffic load and have to work harder than in a sparse network. Hence, the packet dropping rates and end-to-end delay metrics measured in a dense network seem to be worse than these in a spare network. Attack initiated in a dense network will have deviations between benign and adverse node smaller than that of attack initiated in a sparse network [74].

Our approach to study the attack impact in this thesis is to maximise the deviation between the benign and attacked scenario of each metric through setting up ideal network conditions. We choose a small size network with nodes in random position and average traffic. Besides, the nodes are distributed equally, which means the number of neighbours for each node is similar. To discover the trend of the attacks, we initiate them in different positions of the network and find the common changes to the performance metrics. By doing so, it will be easier to detect the sensitive metrics as well as exclude the irrelevant metrics, which are not affected by the attacks. The detailed work to solve this issue will be presented in Chapter 3.

2.6.2. Identifying the Features for the Anomaly-based IDS

Given the review in Section 2.5, building an anomaly-based IDS is the best approach to deal with the performance-type threats. In other networks, there are many features for classifying the attackers' behaviours. For example, in the wired network, a set of 41 features conducted in the KDDCup'99 IDS dataset is utilised commonly in many research efforts [21]. However, the literature on building anomaly-based IDS for 6LoWPAN remains the issue of choosing the right features for reducing the monitored data and effectively detect the attacks.

In order to detect the internal threats in WSN, a number of data features were proposed. Da Silva et al. [75] suggest to monitor the following features:

1. The time between two consecutive messages for detecting the negligence (sending message too slowly) or exhaustion (sending message too quickly)
2. Payload: for discovering integrity attacks, which make changes to the packet payload

3. Delay: detect attacks that make high delay in sending the messages such as Black Hole or Selective Forwarding
4. Repetition: detect DoS attack
5. SenderID: for detecting wormhole, Hello Flood attack – this parameter can also be applied in discovering Neighbour Discovery attacks of IPv6 and Sybil, which create a strange SenderID
6. Number of collisions: detect attacks, which cause large number of collisions such as jamming attacks.

Strikos [76] added the following parameters:

1. Number of lost packets: a higher packet loss rate can be a sign of dropping, modifying or jamming attacks
2. Number of modified packets: this shows the threats of integrity attacks
3. The amount of energies used by the network: this is for preventing the network partition by control the energy consumption distributed in the network.

Each chosen feature only shows its effectiveness in protecting the network from one or some threats but not all. The IDS in 6LoWPAN, however, cannot be implemented to monitor all the parameters for all the threats because of its limited resources. Therefore, the defence system needs to prioritise the threats dependent on the scenario, while choosing only the most important performance metrics that need to be monitored. The chosen metrics should also cooperate in an optimal way to allow the network monitor all these priority attacks.

Our approach to select the features for the anomaly-based IDS is based on the studying of attack impacts in Chapter 3. The detail justifications of the features and their uses are presented in Chapter 4.

2.6.3. Profiling the Benign RPL Behaviours

Anomaly-based IDS seems not to be helpful for detecting topology attacks. This is because malicious nodes may look like normal regarding all the performance metrics. On the other hand, the abnormal topology set up by the attackers cannot be measured through the statistical metrics. As a result, we develop a specification-based IDS using benign behaviours profile of the RPL to detect the topology attacks. Our approach to solve this issue is introduced in Chapter 5.

2.6.4. IDS Placing

Placing the IDS and distributing the monitoring nodes in WSN is other issues to be considered. Common approaches for such issues are the network-based, host-based and the hierarchical deployment.

The network-based approach, which puts the monitoring module on the base station to receive and analyse all the monitored data from the nodes, can utilise the strong resource ability of the base station. Another good thing is that it can use the global view to detect co-operation attacks. However, this architecture creates a lot of communication overhead on intermediate nodes, which need to forward the monitoring data to the base station. Besides, the monitoring data can also be falsified in the middle, which make the architecture bad at detecting the local attacks.

In host-based approach, IDS agents are implemented in every node. Such nodes will monitor, analyse the monitoring data and judge themselves. This method can reduce the monitored traffic but putting more computational work, which will consume node resources and shorten its lifetime. On the other hand, this approach can detect local attacks accurately, but it is lack of a global view for protecting co-operation attacks.

The hierarchical approach [77, 78] combines the two aforementioned placements by clustering the network. IDS agents are put on three levels as shown in Figure 2.8. The first level is the cluster members, which are used to monitor their neighbour behaviours and collect audit data. These nodes can analyse their own collected data to identify malicious neighbours. The second level is the cluster heads, which used as coordinators to aggregate audit data from their cluster nodes, analyse and make decisions to identify the intrusions. The highest level is the base station, which collect the monitoring data from its cluster heads and detects attacks throughout the network.

The main advantages of this architecture lie on detecting distributed attacks and providing scalability. The audit data collected from different views also makes this architecture more robust and fault tolerant [77]. The overhead is created mainly by communication between the clusterhead rather than by monitoring traffic between a clusterhead and its members. Such overhead can be reduced significantly if clusterheads apply new technology such as multiple communication interfaces, to separate the monitoring traffic and the network traffic.

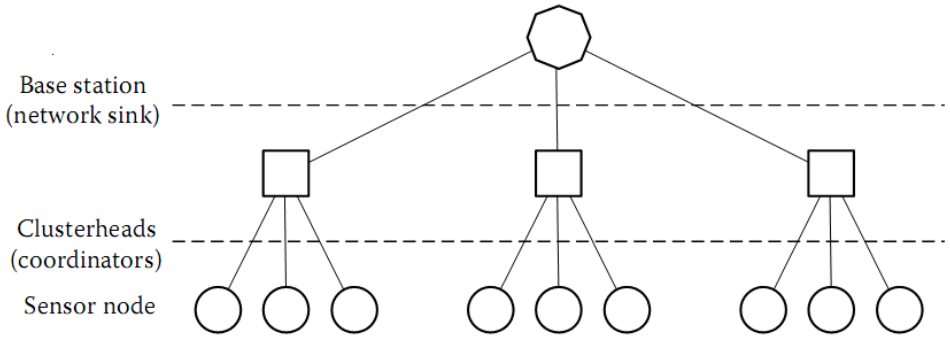


Figure 2.8. Hierarchical approach in putting IDS agents on WSN

The choices of IDS placement depend on the IDS techniques used and the security resource available in the network. In case of 6LoWPAN, a network-based architecture is not suitable for 6LoWPAN security because it will create a huge IDS traffic towards the sink. Host-based IDS is also not feasible because it shorten the host life when overusing its resource for monitoring. Therefore, we propose a hierarchical architecture with the use of additional interface in each monitoring node for deploying IDS that can save resources and extend the IDS capability in this thesis. The detailed architecture will be presented in Chapter 4.

2.6.5. Evaluating the Effectiveness of the Approaches

The literature commonly uses the True and False Positive rate to evaluate the precision of the IDS. The True Positive rate can also be called the Detection Rate as it shows the percentage of malicious behaviours that the IDS can detect. On the other hand, the False Positive rate represents the IDS sensitivity as it shows the percentage of legitimate behaviours, which are detected as intrusions. The formula to calculate these two metrics are:

- **True Positive Rate (TPR):** $TP_{rate} = \frac{TP}{TP+FN}$.
- **False Positive Rate (FPR):** $FP_{rate} = \frac{FP}{FP+TN}$.

Where:

- **True Positive (TP):** is the total correctly detected malicious behaviours. This happens when the IDS correctly raises alert for a malicious event.
- **False Positive (FP):** happens when the IDS erroneously raises a false alarm over a legitimate behaviour in the network.

- **True Negative (TN):** happens when the IDS correctly judges a legitimate behaviour as normal.
- **False Negative (FN):** happens when the IDS erroneously judges a malicious event as normal.

Besides, an efficient IDS should also save the limited node resources, therefore, another evaluation metric is the resource consumption. The more energy and power the network uses, the shorter its lifetime and the IDS effective are. In order to assess the impact to the resource consumption, we utilise the network energy usage and node power consumption as presented in [79]. The formula to calculate these metrics are as follow:

$$\boldsymbol{Energy\ usage\ (mJ)} = (19.5mA * transmit + 21.8mA * listen + 1.8mA * CPU + 0.0545 * LPM) * 3V/4096 * 8 \quad (1)$$

$$\boldsymbol{Power\ Consumption\ (mW)} = \frac{Energy\ usage\ (mJ)}{Time(s)} \quad (2)$$

2.7. Chapter Summary

In this chapter, a thorough review of security threats towards 6LoWPAN was given. We have pointed out that the internal attackers in such network are potentially more dangerous and more significant than the external in terms of downgrading network performance. Internal attackers are also able to initiate a variety of attacks, which make them even more difficult to be detected. Cryptography solutions as the first line of defence are still being developed, but have to overcome big issues of node resource constraint as well as scalability requirement. Those solutions are also not effective when dealing with the internal attackers. Different IDS solutions categorized into anomaly-based and specification-based are reviewed as the second line of defence to protect 6LoWPAN from internal threats. Our review shows that there should be an anomaly-based IDS for detecting the performance-type attacks, while a specification-based IDS integrating RPL profile will be the best approach to detect the topology attacks. Such IDS solutions are still missing in the literature, so filling in this gap will be the main aims of this thesis. We also reviewed the most prominent issues of developing an IDS, including feature selection, RPL profiling, IDS placing, and the parameters for evaluating the effectiveness of the IDS.

Given the two categories of 6LoWPAN internal threats, there is a need to analyse and quantify those attacks to understand their effects to the network performance. The results will be essential to develop the relevant IDS. These are the missions of the next chapter.

CHAPTER 3. BEHAVIOURS OF TYPICAL 6LoWPAN INTERNAL THREATS AND THEIR IMPACTS TO NETWORK PERFORMANCE

3.1. Introduction

As discussed in Section 2.3.5, we categorise the internal threats in 6LoWPAN into two types: the performance-type and the topology-type. In performance attacks, the internal malicious sources focus mainly on manipulating the traffic going through them, the communication channel around, and the operation of other neighbours. For instance, attackers can drop all or part of the packets going through them, or create jam to make the channel around always busy. On the other hand, the topology attacks involve any threats that break the optimal topology of the network, for example, creating loops, attracting traffic, or generating heavy overhead. Internal attackers often combine these two types to increase the impact. For example, they can first use the topology attack to gain the traffics towards a malicious source and then manipulate these traffics through performance attacks applied in this source.

This chapter will quantify the impact of typical attacks in these two categories. We also analyse the key attackers' behaviours to form the basis knowledge for developing the IDS in later chapters.

For the performance-type attack, we choose some prominent and common attacks at physical, MAC and network layer, which manipulate the traffic through the malicious sources (The Black Hole, Grey Hole, and Delaying attack), the channel performance (Jamming attack), or the neighbours (the Hello Flood attack). A short introduction of each attack is given as below:

- **Network/Routing attack:** The malicious node drops all (Black Hole) or part (Grey Hole) of the traffic that passes through it, which causes data loss. The malicious node can gain more impact on network performance by advertising itself as having the shortest route to destination to have more traffic from the neighbours.

- **Delaying attack:** the malicious node stops forwarding the packets passing through it, making the other receiver wait and thus add delay to the transmission [80].
- **Physical level attack:** A malicious node or other device purposefully tries to interfere with physical transmission and reception of wireless communication. The authors of [81] and [82] distinguish four types of a jammer. (i) a constant jammer continually emits the radio channel even when it is busy (ii) attackers continuously broadcast dump packet (iii) jammer alternates between sleeping and jamming randomly, and (iv) jammer stays quiet when the channel is idle but starts transmitting a radio signal as soon as it senses activity on the channel. For simplicity, in this thesis, we will consider only the first two types of this attack.
- **Signalling:** uses laptop class or high power device to send the Hello messages to the whole network to make other nodes confused about unreachable neighbours.

Based on the analysis of these four typical attacks, we also discuss the similarity of the behaviours of other attacks, including the selective forwarding, the exhaustion, the collision, the Sybil and Clone ID, and the Wormhole attack.

For the topology attack type, we discuss some particular variations including the Sinkhole, the Rank, the Local repair, and the DIS attack. Most of these attacks have been discussed in our previous work [8-10, 83]. Their attack mechanisms are summarised as follows.

- **The Sinkhole attack**: the malicious source will propagate its rank with a good value, normally the same rank of the sink. As a result, its neighbours will select it as their preferred parent and send traffic to that node. The Sinkhole attack is often combined with other traffic-attacks to manipulate the traffic it attracted.
- **The Rank attack:** after the attack is triggered, the malicious source changes the way it processes the DIO messages from other neighbours so that it will get the node with the worst rank as the preferred parent. This kind of attack will create un-optimised route for all the packets that go through the malicious source.
- **The Local repair attack:** after the attack is triggered, the malicious node starts broadcasting local repair messages periodically, though there is no problem with the link quality around the node. Other nodes upon receiving the local repair messages will need to recalculate the route that is related to the malicious nodes.

This kind of attack creates more control overhead messages, as well as some packet dropping because of temporarily unavailable route.

- **The DIS attack:** after the attack is triggered, the malicious nodes will send the DIS messages periodically to its neighbours. The DIS messages can be sent in two ways, which will lead to a different response from the receivers. The first way is to broadcast the DISs to make receivers reset their DIO timer as the broadcast DISs mean that the topology around is changed. The second way is to unicast this DIS message to all nodes in the neighbour list, the receivers upon receive will unicast DIO message to the sender. Both of these ways add more control overhead on the network.

In this chapter, we first present the metrics to evaluate the attack impact and the general evaluation scenarios. We then go more detail into studying of the behaviours of the aforementioned attacks. For each attack, we will go through the three steps:

1. Implementation of this attack in Cooja-Contiki, a well-known simulation and real time platform designed for 6LoWPAN [84];
2. Discuss the simulation results of the evaluation scenarios;
3. Obtain the specific attack behaviours.

The study of those attacks in this chapter will be fundamental for developing the IDS module in the next chapters.

3.2. Metrics that Represented Attack Impact and Node Behaviours

To understand the attack impact, we select the three most sensitive and relevant metrics, which are as follows.

- **Packet delivery ratio** is calculated by the percentage of the number of packets received per number of packets sent. The attacks targeting adding collisions and packet drop, which will lead to significant change to this metric.
- **End-to-end delay** is the average time that a message needs to transfer from the source to the sink. This metric is sensitive when the network is under collision and delay attacks. An observed low end-to-end delay does not always imply a good network performance. For example, too many packets dropped can make

communication on the radio channel faster, which also leads to a low end-to-end delay.

- **Routing control overhead** is the total number of DIO, DAO, DIS messages generated by all the nodes during network run time. One of the most important goals of 6LoWPAN design is to minimise the routing control overhead to save communication and computational resources. Therefore, an abnormal increase in routing control overhead may indicate an attack on topology.

On the other hand, to understand the changes on the node behaviours, we track its local parameters regarding both protocol operation and communication performance. At each node, we measure the *number of control messages generated* and the *change in rank* metrics to reflect the RPL behaviours. As discussed in Section 2.2, these metrics reflect the stability of the network. For the communication, we choose varied local metrics from different layers to reflect the potential changes in performance behaviours. These metrics include *forwarding rate, forwarding delay, power consumption, packet collision ratio, and signal strength*. A summary of the metric definitions is given below.

- **Number of control messages generate:** we will consider the number of DIO, DAO, and DIS messages generated by a single node during the simulation time
- **Change in rank:** the rank of the nodes through time will be recorded to measure how many times it changes
- **Forwarding rate:** the number of packets that are forwarded per the number of packets that are required to be forwarded in a period of time
- **Forwarding delay:** average time between receiving a packets and forwarding this packet to the next neighbour
- **Power consumption rate:** the power that a node consumes after a period of time
- **Packet collision ratio:** the number of collisions reported by the nodes through time
- **Signal strength:** the signal strength of the node through time, which represented by the RSSI metric at the receiver

In order to see the impact of the attacks better, we also divide the simulation time in multiple fixed-size periods, called window times. Monitoring the trends of the metrics through the window times will allow us to understand the attack impacts on network performance and node behaviours better. For each attack, we will report metrics that are observed to have significant change when comparing the normal to the malicious source.

3.3. Setting Scenarios for Evaluation

As discussed in Section 2.6.1, we choose a small size network with ideal working conditions to maximise the deviation between the benign and attacked scenarios to detect the sensitive metrics more easily. Our simulations were conducted with Coola Contiki-2.6, a simulation tool environment that provides support for RPL. Our simulated network consists of 25 fixed nodes placed randomly in a 250x250m area; each node has a transmission range of 50m. In these 25 nodes, there is a sink placed near the top left corner and 24 senders placed randomly around the sink. Every sender sends packets to the sink at the rate of one packet every 15 seconds.

It is well known in WSN that the location of the attackers will create different impacts on network performance, i.e. impact of the attackers located near the sink will be more severe than that of the attackers located near the border. Therefore, we will evaluate the attack impact in different locations of the network. For comparison, in each attack, we compromised the nodes with the same locations to eliminate the impact of the attack location. Every simulation was repeated 10 times using 10 different random seed; the results were averaged over those runs to be more accurate.

We first set up a normal scenario with no attacker and normalize the simulation results for comparing with the compromised scenario in the later stage. We choose some random nodes to implement the particular attack code and invoke it at a specific time. This specific time is set to be 60 seconds after the network starts so that the network is in its stable condition. Each simulation scenario is run for 330 seconds, but the senders only send 30 packets (so the sending stops after around 300 seconds). The reason for doing that is to eliminate any effect on performance results from the uncompleted transmission. The network performance for the compromised scenarios are then compared with those in the standard scenarios to reveal the impact of each attack.

Figure 3.1 illustrates the network topology setup in our scenarios. In this figure, the green node (marked number 1) is the Sink; the other yellow nodes are the senders, which send messages towards the Sink. In the compromised scenarios, some of the yellow nodes will change the operation code to become the internal attackers. Figure 3.2 shows the connectivity of the network in Figure 3.1. The common parameters used in these simulations are summarised in Table 3.1.

In our previous work which quantifying 6LoWPAN attack impacts [9, 10], we implement an extensive simulation of attackers manipulated in every position of the network. The results of these works point out that positions which are near the sink and/or having many child nodes are the locations that attackers can create the most impact. This observation is also in line with common-knowledge in the literature. Hence, in this thesis, we only initiate the attacks at positions with such characteristics to study the maximum effects that attacks can create to network performance. The impact of the node performance will depend on the load that it needs to forward. If a high traffic node is compromised, the overall performance of the network will be downgraded significantly. Therefore, attacks on nodes with similar forwarding load tend to have similar impact on network performance. As nodes forward packets from the other nodes towards the sink, the nearer the node to the sink, the more load it will have. When nodes have the same number of hops towards the sink, in as equally distributed network, their forwarding loads will be similar. Hence, the experiments can focus only on several typical nodes that represent different path length towards the sink, rather than focus on all the nodes. For the experiment in this chapter, those typical nodes chosen to implement the attacks are the five nodes. As can be seen from the topology figure, those nodes form a backbone to forward packets from other nodes, with different number of hops towards the sink. Impact of attack on other nodes can be referred to impact of those typical nodes, given the corresponding path length. The chosen five positions are node 6, 22, 8, 9, 10. The improvements to compare with the work in [9, 10] is that we add several node performance features (listed in Section 3.2) to study the attacker behaviours in more detail.

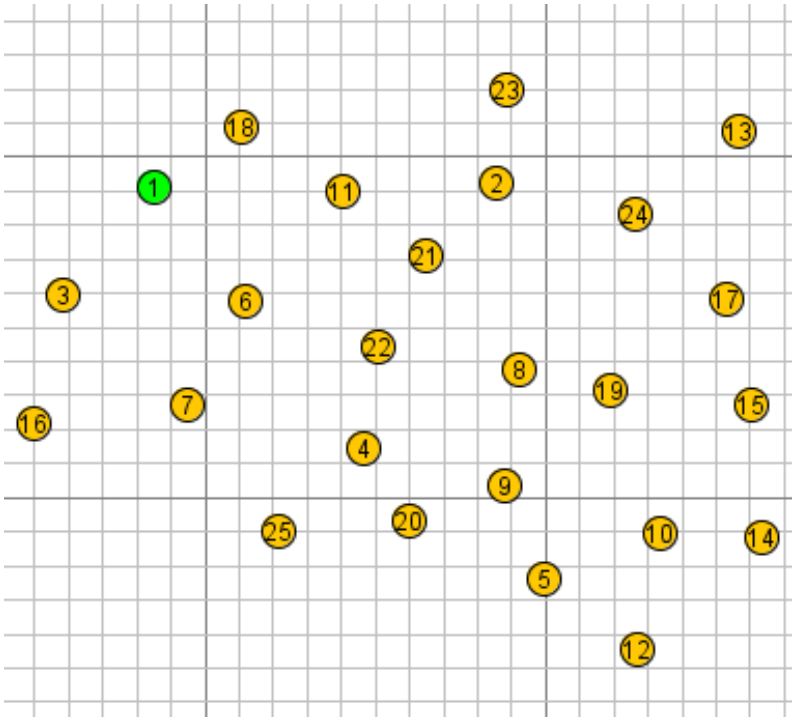


Figure 3.1. Network scenario set up

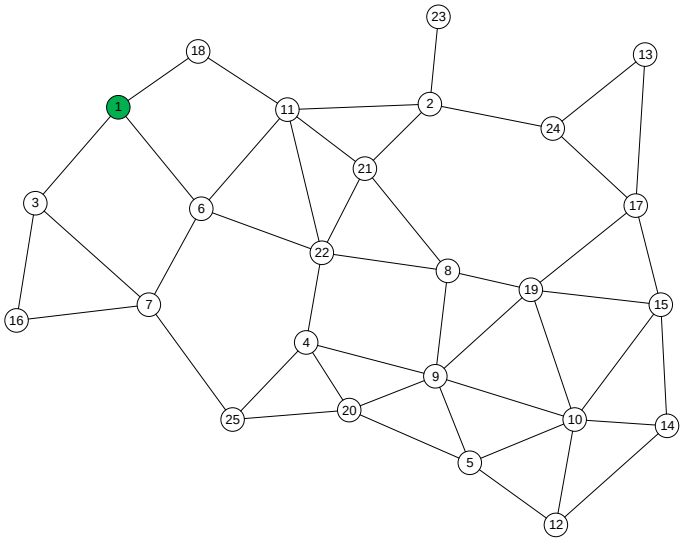


Figure 3.2. Connectivity map

Table 3.1. Common simulation parameters

Parameters	Value
Simulation Platform	Cooja Contiki 2.6
Number of nodes	24 senders and 1 sink
Traffic model	Constant bit rate
Sending rate	1 packet every 15 seconds
Simulation run time	330 seconds
Number of sending packets per each sender	20 packets

3.4. Assessing Typical Performance-type Internal Threats

Overall, there are three major ways to downgrade the network performance directly through the malicious sources. First, the attackers can manipulate the traffic going through them, for example, dropping, delaying, or adding more traffic to attacking the next hop in the forwarding flow. By doing so, the attackers can affect network performance in multiple aspects, for instance, they can lower the forwarding rate, or increase the forwarding delay, and power consumption. Second, the attackers can also manipulate the channel by creating more collisions to prevent the communication. This type of behaviours will make the neighbours send packets more slowly because they need to wait for the channel to be free, or need to resend the packets after the collisions, not to mention the power waste for such activities. Finally, the attacker can create fake ID or send fraudulent messages to make the neighbours think that there are new nodes around. The neighbours will then waste more resource and perform less effectively because of processing this information, which later downgrades the network performance in general. Within the scope of this thesis, we only select several typical attacks to assess. Those attacks need to represent the three aforementioned attack methods, while showing clear behaviours for better understanding the attack mechanisms.

In detail, for the traffic attack type, we choose the Black Hole/Grey Hole and the Delaying attack. The Jamming attack is selected for understanding the abnormal channel behaviours, while the Hello Flood attack is picked to represent the abnormal neighbour activities. Assessments of such attacks are given in detail in the following sections.

We will also discuss other attacks' behaviours based on the analysis of common characteristics with the typical attacks at the end of this section.

3.4.1. Black Hole/Grey Hole Attack

The Black Hole attackers will drop all the packets that are forwarded through them. To increase the attack impact, attackers first try to advertise itself as providing the best route to the destination to attract as much traffic as possible (this can be considered as combining with other attacks, for instance, Sinkhole). After getting the essential traffic, it will start the packet dropping.

The Grey Hole attack can be considered a different form of the Black Hole attack. The difference is that instead of dropping all of the packets, the Grey Hole attackers select

only part of the traffic through it to drop. By doing so, the Grey Hole attackers can still gain the trust from the neighbours and bypass the justification of having bad behaviours.

To assess the impact of the Black Hole/Grey Hole as a single attack, we only implement the dropping behaviour after the attackers attract essential traffics.

3.4.1.1. Attack implementation

The implementations of these two types of attacks are given in the pseudo code as in Figure 3.3 below.

```
//Start the Blackhole attack after t second
1: on_receving_packets_to_forward {
2: if (time < t)
3:         {operate_like_normal;}
4: else  {drop_packets; }
5: }
```

```
//Start the Grey Hole attack after t second
1: on_receving_packets_to_forward {
2: if (time < t)
3:         {operate_like_normal;}
4: else  {
5:         if (dropping_condition) {drop_packets; }
6: Else {operate_like_normal;} }
7: }
```

Figure 3.3. Pseudo code of Black Hole and Grey Hole attacks

The dropping conditions for the Grey Hole attack can be varied with at least two options as follows:

- Dropping based on time, e.g. forwarding all the packets that come in the first half of a period and dropping all the packets that come in the remaining time of such period
- Dropping based on node ID, e.g. forward all the packets that come to the odd ID nodes but drop all the packets that come to the even ID nodes.

In this particular simulation, we set the Grey Hole node to drop 20% of the packets that passed through it using time as the condition.

3.4.1.2. Impacts on Network Performance

Table 3.2 and 3.3 shows the simulation results of the Black Hole and Grey Hole attack, respectively, to compare with the normal network performance scenario. As can be seen from these tables, the average end-to-end delay, the total overhead, total collisions, and total rank change metrics in the attack scenarios are similar to that of the normal scenario. On the other hand, the global delivery ratio is changed significantly, but only when the attacks are initiated at node 6, 22, and 8. Figure 3.4 depicts such changes in more detail. As can be seen from the figure, the Black Hole/Grey Hole attacks create more packet loss

when they are implemented nearer to the sink. Performance logged of local forwarding over the whole network shows that the positions help the nodes to attract traffic, for example, node 6, 22, and 8 in Black Hole attack scenario has a traffic of 259, 158, and 115 packets that need to be forwarded through it, respectively. Because those nodes drop most of these packets, the global delivery ratio decreases significantly. A similar phenomenon was also observed with the Grey Hole attack.

In case of initiating at node 9 and node 10, the Black Hole/Grey Hole attackers do not affect to the network performance because such positions are near the border of the network, and the neighbours of such nodes also have other routing option to avoid the dropping route.

Table 3.2. Performance comparisons between the normal and Black Hole attack scenarios

Scenario	Global Delivery ratio	Average End-to-end delay (ms)	Total Overhead	Total collisions	Total Rank changes
Normal	1	531.27	371	98	111
Att at 6	0.53	495.33	375	125	110
Att at 22	0.71	449.31	381	99	112
Att at 8	0.79	439.16	378	85	112
Att at 10	1	538.57	374	98	111
Att at 9	1	551.27	378	98	111

Table 3.3. Performance comparisons between the normal and Grey Hole attack scenarios

Scenario	Global Delivery ratio	Average End-to-end delay (ms)	Total Overhead	Total collisions	Total Rank changes
Normal	1	531.27	371	98	111
Att at 6	0.78	500.36	369	98	115
Att at 22	0.88	487.75	378	97	113
Att at 8	0.91	487.15	377	98	103
Att at 10	1	539.67	375	98	111
Att at 9	1	534.26	378	98	111

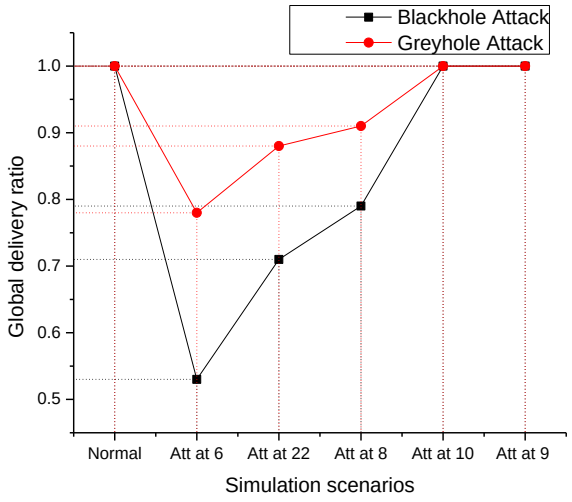


Figure 3.4. Global delivery ratio comparisons between the normal and Black Hole/Grey Hole attack scenarios

3.4.1.3. Attack Behaviours

We can conclude some important points from the simulations as follow. First, the 6LoWPAN RPL does not have any effective mechanism to detect or react to the Black Hole/Grey Hole attacks. From the protocol point of view, the dropping behaviour of the forwarder is silent to the sender. The senders only check the ACK messages from the forwarder. Once received such ACK, they assume the forwarder got the packets and will do the remaining forwarding. As such, the forwarders can drop the packets without being detected. Second, the metrics that are sensitive to the Black Hole/Grey Hole attack are all the metrics related to the local forwarding process, including the local forwarding traffic, and the local forwarding rate. Both of these metrics need to be considered in judgement because the use of a single metric does not always give correct result. For example, considering when a node is asked only to forward two packets during the simulation time, and it drops one packet, the forwarding rate in this case will be 0.5, which is very low, but it is unlikely that this node is the Grey Hole attacker. However, if significant traffic is observed in a node, but its forwarding rate is significantly low, there will be a high chance of the Black Hole/Grey Hole attackers.

3.4.2. Delaying attack

In the delaying attack, the malicious nodes will stop the traffic, which goes through it for a while after forwarding. This attack threatens the particular sensor network application, which require real time traffic. The delaying behaviours help the attackers still gain the credit of forwarding packets, yet making the neighbours have less opportunities to

forward data to others due to run out of waiting time. The main objective of this attack is to create traffic disruption.

3.4.2.1. Attack Implementation

The implementation of the delaying attack is given in the pseudo code below. In detail, the attackers start the attack after t seconds waiting on the network. Since then, for every traffic going through them, they will be waiting for a pre-set time before forwarding the packets. In our implementation, we choose the waiting time around 2 seconds. The Pseudo code for implementing the delaying attack is given as in Figure 3.5 below.

```
//Delaying attack after t seconds
1: on_receving_packets_to_forwards {
2:        if (time > t) {
3:                waiting_for_an_attack_period;
4:                forward_packets; } else {
5:                        forward_packets}
6:        }}
```

Figure 3.5. Pseudo code implementation of delaying attack

3.4.2.2. Impacts on Network Performance

Table 3.4 shows the simulation results of the delaying attack to compare with the normal network performance scenario. As can be seen from the table, the attack does not create significant impacts when it sits near the border of the network, e.g. node 9, 10. The nearer to the sink the attacker is, the more severe impact it can cause to the network performance. The average delay has significant change, which is up to 150%, while the total overhead, collisions and rank changes measured in the network are even increased more, from around 150% to more than 200%. On the other hand, the global delivery ratio shows a slightly decrease between 94 to 96%. The consequences of these changes lead to more energy consumed, with up to 10% increase in computational power, and up to 31% wasting more in communication energy. Comparisons of the energy consumed in these scenarios are illustrated in Figure 3.6.

Table 3.4. Performance comparisons between the normal and delaying attack scenarios

Scenario	Global Delivery ratio	Average End-to-end delay (ms)	Total Overhead	Total collisions	Total Rank changes
Normal	1	531.27	371	98	111
Att at 10	1	544.35	376	102	111
Att at 9	1	539.27	398	98	113
Att at 22	0.96	569.29	546	153	176
Att at 8	0.95	696.95	507	206	179
Att at 6	0.94	784.77	602	215	162

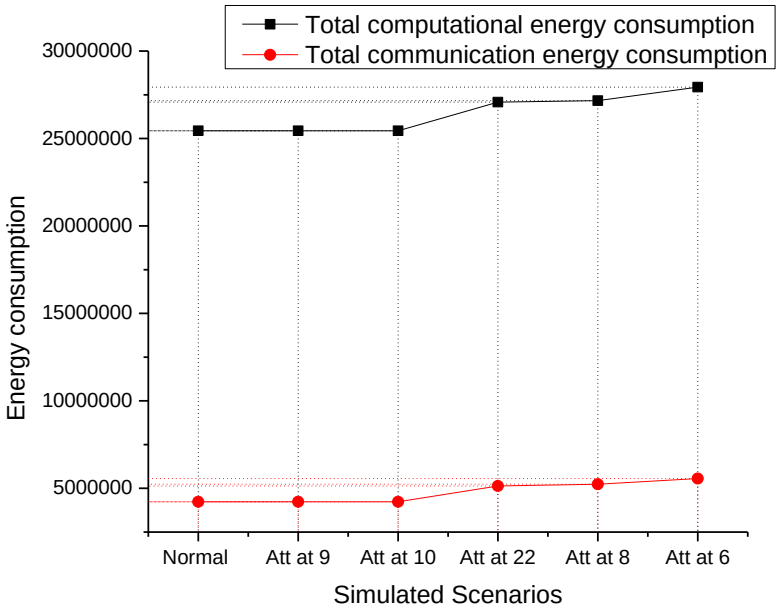


Figure 3.6. Comparison of the computational energy consumption between the normal and delaying attack scenarios

Local performance logged data also reveals that most of the packet dropping was caused by the attacker nodes. The reason is that the forwarding packets were kept too long in the malicious nodes' limited size buffer. These packets will be dropped once the incoming packets go over the buffer size.

3.4.2.3. Attack Behaviours

When monitoring the attackers' local behaviours, we see clearly the abnormal increase of forwarding delay and packet dropping due to the limited buffer size. Besides, the rank metric was updated more often, which leads to the increase of number of DIO messages, from two to three times compare with the normal scenarios. This is because the RPL uses the Expected Transmission Count (ETX) as the main metric when calculating the node rank, and due to unpredicted forwarding/dropping packet behaviour, this metric changed

frequently. This change is reflected through the DIO messages sent from the attackers. Consequently, the nodes around this malicious node also need to recalculate and update its rank. As a result, there will be overhead and collisions generated in the network.

3.4.3. Jamming attack

3.4.3.1. Attack Implementation

The main goal of the jamming attacker is to block the communications nearby by disrupting the neighbours' ability to transmit or receive packets. There are different ways to jam the network, for example, using the high power transmission to dominate the transmission of other surrounding nodes, or keep making the communication channel busy to prevent neighbours sending/receiving packets of the neighbours. Most of the literature regarding jamming attacks have followed the classical classification of Xu et al. [81], in which the authors divided the attack into 4 main following types:

1. **Type I - Constant jammers:** This jammer transmits the messages constantly without following the MAC protocol, for example, sending the messages even when hearing busy signal in the channel.

2. **Type II - Deceptive jammers:** This jammer constantly injects dump packets into the network to keep other nodes to remain in the receiving state.

3. **Type III - Random jammers:** This can be considered as an energy efficient attack for jammers that have limited power supply. The jammer randomly chooses a period of time to sleep and a random period to jam. When the jammer is in the jam state, it can perform either constant or deceptive jamming.

4. **Type IV - Reactive jammers:** The previous types of jammers are considered as active jammers, which attack regardless of the communication state of the victim nodes. In reactive jamming, jammers will remain silent and will only jam when they sense valid traffic being exchanged in the network. This type of jammers is harder to detect compared to the active jammer types.

The common behaviours of these four types are to send the signal or packet without caring about the availability of the channel, to create more collisions and make the channel busy as long as possible. The difference between these types is that they choose the time to start the attack to save the resource and make it difficult for the defender to detect. As our main concern is to study the attacks' behaviours, for simplicity, we only choose type I (J-I) and type II (J-II) of jamming attack to implement in 6LoWPAN Contiki. The implementation makes the jamming attacker send packets without caring about the state of the channel (J-I), and broadcast a dump packet periodically after initiating (J-II). The Pseudo code for this implementation is given in Figure 3.7 below.

```
//J-I after t seconds
1: if (time > t) {
2:        is_channel_busy = FALSE;
//Will send packets even in busy channel
3: }
```

```
//J-II after t seconds
1: if (time > t) {
2:        time_count = 0;
3:        while (counting_time)  {
4:        if (time_count > period) {
5:                broadcast(dump_packet);
6:                time_count = 0;}
7:        }}
```

Figure 3.7. Implementation of two types of Jamming attack J-I and J-II

3.4.3.2. Impacts on Network Performance

Table 3.5 and 3.6 show the simulation results of the Jamming attacks to compare with the normal network performance scenario. These tables indicate that the average end-to-end delay and the total collisions have the most significant changes. In J-II, the average end-to-end delay increase from 640ms to 760ms, about 120%-140% of the normal scenario. The total collisions increase even more, with about 165%-268% to compare to the normal scenario value. Total overhead and total rank changes increase slightly, which indicates that the optimal topology is affected insignificantly, while the high global delivery ratios imply only a small amount of packets are dropped. Given that a jamming attacker will create many collisions, the sending packets of the surrounding nodes will be delayed, or need to be resent after the collisions. Therefore, end-to-end delay will be increased. When looking in more detail at the collisions, we observed that the centre of the network, i.e. an area consisting of node 4, 22, 21, 8, 19, 9, and 20, has the most collisions regardless of the attacking positions. Hence, we can conclude that the jamming attack does not only affect to the performance of the surrounding nodes, but it also spreads the impact on a

larger area, because its neighbours in their turns will continue to affect the behaviours of their surrounding nodes.

Table 3.5. Performance comparisons between the normal and J-I scenarios

Scenario	Global Delivery ratio	Average End-to-end delay (ms)	Total Overhead	Total collisions	Total Rank changes
Normal	1	531.27	371	98	111
Att at 9	1	528.11	381	117	113
Att at 10	1	559.95	366	130	111
Att at 22	0.98	569.35	387	126	120
Att at 6	0.97	579.2	412	196	119
Att at 8	0.99	607.88	407	147	117

Table 3.6. Performance comparisons between the normal and J-II scenarios

Scenario	Global Delivery ratio	Average End-to-end delay (ms)	Total Overhead	Total collisions	Total Rank changes
Normal	1	531.27	371	98	111
Att at 8	0.99	639.18	389	214	123
Att at 10	0.99	655.27	395	195	135
Att at 22	1	733.62	399	201	130
Att at 6	1	759.93	369	162	117
Att at 9	0.99	760.06	410	263	137

As the collisions will lead to packet retransmissions, an additional overhead regarding the communication and computational consumption is also observed. Figure 3.8 and 3.9 indicate the comparisons of the computational and communicational energy consumption between the normal and jamming attack scenarios. The figures clearly show the additional overhead created, which is up to 5% in J-I and 8% in J-II in case of CPU usage, and up to 17.4% in type 1 and 24.4% in type 2 in the case of transmitting handling. These will significantly decrease the node lifetime, given their resource constraint nature.

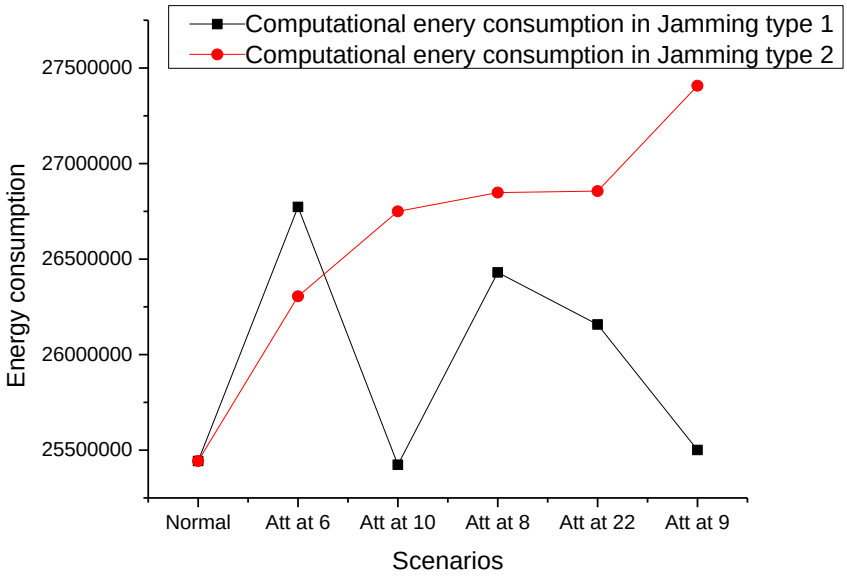


Figure 3.8. Comparison of the computational energy consumption between the normal and 2 types of jamming attack scenarios

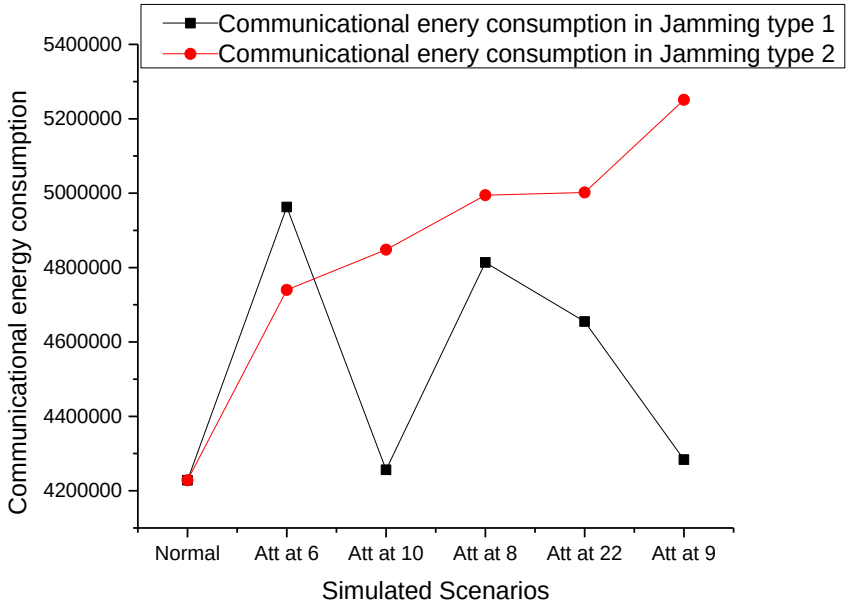


Figure 3.9. Comparison of the communication energy consumption between the normal and two types of jamming attack scenarios

3.4.3.3. Attack Behaviours

The main metric to identify the jamming attackers' behaviour is the collision that this node creates. In J-I, the more traffic that goes through the attacker's area, the more collisions it will cause. On the other hand, in case of J-II, the attacker creates the maximum collisions not when it sits in the high traffic area, but when it has more connections with nodes that are in the high traffic area. This is because the constant

dumping of sent packets in that case will interfere with the performance of those high-traffic nodes, and add more collisions in more areas rather than just affecting a single area. The power consumption metric is sensitive, especially in J-II, because the attackers operate constantly, so it will consume significantly more energy compared with that in the normal case.

3.4.4. Hello-flood

In some routing protocols for wireless ad hoc and WSN, the Hello messages are used to announce themselves to their neighbours. A node, which receives such a message, may assume that it is within a radio range of the sender and take into account of this sender when selecting the route. In case of 6LoWPAN RPL, the DIO messages play the role of such a Hello packet.

Hello Flood attackers can utilise its large transmission power (for example, using the laptop when pretending to be an internal node) to broadcast these Hello messages to all over the network. Such behaviour could convince every other node in the network that the attacker is its neighbour. The consequence is that, if the attacker advertises a high quality routing information, it will attract traffic from many nodes in the network. However, such nodes cannot send their packets to the advertiser due to the limit in transmission power and they will be confused about the communication. Because exchanging the neighbour information is very important in maintaining the optimal topology in RPL, the Hello-DIO messages may create significant impact on the network performance.

3.4.4.1. Attack Implementation

The implementation of the Hello Flood attack was first introduced in [85]. In that work, the authors implemented the HELLO flood by letting a malicious node have the ability to send data to all other nodes in the network, but only nodes physically close to the attacker can respond. Our implementation follows this principle, however, we focus more on the attack impact and the behaviours of the attackers, rather than just observing the protocol reaction as in [85]. For implementation, the signal strength of the attacking node is also set to -10dBm, a considerably larger than normal signal strength of the sensor nodes (from -50 to -70 dBm) to send the signal over the network. Besides, the attackers are set to have direct connection to all the nodes in the network. An illustration of the Hello Flood implementation is given in Figure 3.10 below.

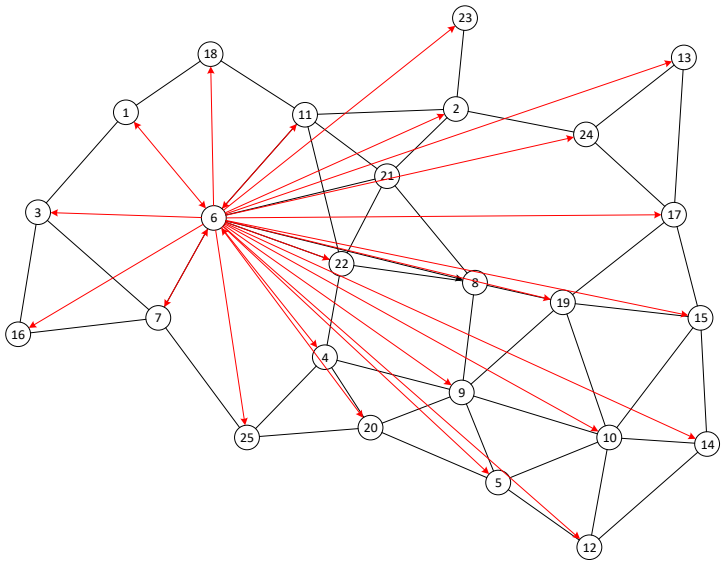


Figure 3.10. Implementation of the Hello flood attack on node 6

3.4.4.2. Impacts on Network Performance

Table 3.7 shows the simulation results of the Hello Flood attack to compare with the normal network performance scenario. As can be seen from this table, the average end-to-end delay, the total overhead, the total collisions and the total rank changes are the metrics that have significant changes under this attack. On the other hand, the global delivery ratio seems to have no change, which indicate that the benign nodes can still route the packets correctly to the sink. Our observation on the detailed communication shows that there is no traffic going through the attacker node, even when this attacker sit near the sink, e.g. node 6. The reason is not only because of the fake route that the senders cannot reach, but also due to the strong signal of the attackers that prevents the communication. Because we do not implement other mechanism to advertise the routing information, the attacker will use the benign rank information. Therefore, the nearer the attacker to the sink, the higher rank value it will have, and this will attract more rank change from other nodes. This explains the significant increase in the number of rank changes, collisions and overhead when this attack is initiated in node 6. However, the most severe impact of the attack happens when it is implemented at node 8, which has worse rank than node 6. The potential reason for this phenomenon is that node 8 sitting near the centre of the network, so its strong signal strength will affect a wider range of nodes. For example, node 6's distances toward the border nodes like 13, 17, 15, 14, 12 are much further than the furthest distance of node 8 to other nodes, hence it will not have as strong impact as node 8.

Table 3.7. Performance comparisons between the normal and Hello flood attack scenarios

Scenario	Global Delivery ratio	Average End-to-end delay (ms)	Total Overhead	Total collisions	Total Rank changes
Normal	1	531.27	371	98	111
Att at 22	0.98	421.99	429	315	108
Att at 9	0.99	456.48	409	116	102
Att at 10	1	510.55	395	67	101
Att at 6	0.98	547.11	511	341	143
Att at 8	0.96	685.61	1022	127	361

3.4.4.3. Attack Behaviours

The prominent characteristic of the Hello Flood attackers is that they have an abnormal signal strength, which is out of range of the normal sensor device. This characteristic cannot hide because the attackers need such a strong signal strength to send the packets to far distances. Another important observation is that the attacker may advertise high routing information value to attract traffic, but from the simulation, even in these cases, there is only limited traffic that choose to go through this node. Apart from those behaviours, there is not much significant evidence to detect this attack from other metrics. This is because the limited traffic goes through the attackers make them almost silent under the monitoring of the communication, so they look like having similar behaviours to other benign leaf nodes.

3.4.5. Other Similar Attack Considerations

This section analyses several common attacks that are similar to the afore-discussed attacks, including the selective forwarding, the exhaustion, the collision attack, the Sybil attack, and the Wormhole attack.

The selective forwarding: this attack was first described by Karlof and Wagner [86]. In a simple form of selective forwarding attack, malicious nodes try to stop the packets in the network by refusing to forward or drop the messages passing through them. There are some different forms of selective forwarding attack. In one form of the selective forwarding attack, the malicious node can selectively drop the packets coming from a particular node or a group of nodes. This behaviour will cause a DoS attack for nodes that need those nodes to forward the packets. We have assessed similar behaviour in the Grey Hole attack in Section 3.4.1. In another form, the selective forwarding can also behave like a Black Hole attacker, in which it refuses to forward every packet except the control

messages. With such behaviours, the sensitive metrics to this attack are the local forwarding load and the local forwarding rate.

The exhaustion attack: this attack is launched by the adversary from multiple ends of the network with the intent of exhausting the limited energy resources of the victim nodes [87]. As a result of the attack, target nodes are overwhelmed with higher than normal intensities of traffic inflow, that will lead to the rapid exhaustion of their limited energy resources; incapacitating them from further participation in crucial network operations. There are several ways to implement the exhaustion attack, for example, requiring the victims to send the CTS acknowledgement constantly by sending continuously the RTS packet [88]; or using strong power source to interfere with the channel [87]. These behaviours are similar with the behaviours of the Jamming and Hello flood, so the sensitive metrics to this attack are the local collision rate and the nodes' RSSI.

The collision attack: According to the author of [88], the collision attack happens when the adversaries sending out some packets to disrupt the current communication of the victims. The MAC protocol in every network is responsible for controlling the media channel by arranging the order of the nodes sending packets, therefore, if a node act selfishly without caring about the state of the channel, it will easily disrupt the communication. This attack's behaviour is similar to the Jamming attack type 1 as discussed before. The sensitive metric to monitor is the local collision rate.

The Sybil attack and the Clone ID attack: The concept of Sybil (or multiple-identity) attacks was first proposed by Douceur in P2P networks [89], and it is defined as a single node has multiple identities to disrupt the accordance among the entities and physical devices in a network. A malicious node forges multiple identities to mislead the network and let the neighbour nodes to believe that they have several trusted neighbours [90]. In a clone ID attack, which is similar to the Sybil attack, an attacker copies the identities of a valid node onto another physical node. This can, for example, be used in order to gain access to a larger part of the network or in order to overcome voting schemes [85]. These two types of attacks aim at creating illegitimate neighbours to confuse the victims' communication, just like the behaviour of the Hello Flood attack. Hence, the RSSI metric is sensitive to this kind of attack as well. For example, there is a very low probability for two nodes in different location with different resource to have the same RSSI, so the difference on RSSI between the nodes can be used to detect the Sybil attack. Similarly, it is also impossible that a node has two different RSSI values, given that it stays at the same

place, so the difference on RSSI through time of a node will help to detect the Clone ID attack. Other metrics to consider are the advertise rank and the traffic. The reason of using these metrics is that whenever we observe that nodes, which advertise good rank but having only little traffic, those nodes will need to be suspected.

The Neighbour attack: After the attack is triggered, the malicious node will replicate any DIO messages that it receives and broadcast them again [10]. The victims who receive this type of message may think that it has a new neighbour, which is not in range. Moreover, if the new neighbour advertises a good rank then the victims may request it as the preferred parent and change the route to the out range neighbours. This attack is also similar to the Sybil, Clone ID, and Hello Flood attack. The behaviours of these attackers can also be detected through the abnormal signal strength, as well as abnormal traffic regarding the advertise routing information.

The Wormhole attack: A wormhole is an out of band connection between two nodes using wired or wireless links. Wormholes can be used to forward packets faster than via normal paths. A wormhole in itself is not necessarily a security breach; for example, a wormhole can be used to forward mission critical messages where high throughput is important, and the rest of the traffic follows the normal path [85]. Therefore, the wormhole attacker normally combines with other attacks for a more serious security threat. The behaviours of the wormhole alone are legitimate to the network, therefore, the detection sign of this attack is usually the detection signs of other attacks that were launched based on the Wormhole attack.

Apart from these performance-type attacks, there is another type that not aiming directly at either the traffic, channel, or neighbours, which is the topology attacks. Unlike the node performance attack, we will not see any straight impact on the nodes' local performance, the communication channel or the anomaly in the signal strength. Instead of that, the internal attackers change the protocol behaviours to break the optimal network topology. As a result, with an un-optimal topology, even when the network nodes perform to their maximum capability, the network performance is still downgrade. We will consider attacks falling in this category in the next section.

3.5. Assessing Typical Topology-type Internal Threats

Because the 6LoWPAN uses the RPL as its underlying routing protocol, we will only assess topology attacks, which aim specifically at RPL. The most common topology

attack in every sensor network routing protocol is the Sinkhole attack, which can also be applied to RPL, so we will first consider this attack. The remaining RPL attacks are selected from our work [8-10, 83], including the Rank attack, the Local Repair attack, and the DIS attack. For each attack, we will follow the similar structure as in Section 3.4, in which we first present the attack implementation, assess the impact of the attack to the network performance, before obtaining specific characteristics of the attackers.

3.5.1. The Sinkhole Attack

In Sinkhole attacks [86], a malicious node advertises an artificial beneficial routing path and attracts many nearby nodes to route traffic through it. An attacker can launch a Sinkhole by advertising a better rank thus attracting nodes down in the DODAG to select it as the parent. RPL, however, uses the link-layer quality to calculate routes, which make Sinkhole, attack less effective in RPL-based networks.

3.5.1.1. Attack Implementation

The implementation of RPL Sinkhole attack was introduced in [85]. In that work, the authors implemented the attack by changing the advertised rank through the DIO messages. Any delay normally used to reduce network congestion is also removed in order to allow the malicious node to be the first node to advertise such a beneficial route. We will follow this way of implementation; however, we will focus mainly on the impact of this attack on the network performance and the behaviours of the attackers, rather than just observing the protocol reaction.

3.5.1.2. Impacts on Network Performance

Table 3.8 shows the simulation results of the RPL Sinkhole attack to compare with the normal network performance scenario. As can be seen from the table, it is interesting that the Sinkhole attack started at the border of the network have a lot more impact on the network performance to compare with those started near the sink, which is unlike any other attacks we have seen. The reason is simply that Sinkhole attackers in the network border will attract the traffic to go far from the sink, hence creating more dropping rate, delay and overhead. In case the Sinkhole attacker sits near the sink, because this node does not do any other attacking mechanism, it will forward the packet to the real sink, so the network performance does not downgrade a lot. The low average end-to-end delay observations in the cases where the attackers sit at node 9 and node 10 do not mean that

the attacked network transfers data faster than the normal network. Given that these two cases have a very low global delivery ratio, many packets were dropped. We do not count those dropped messages in the average end-to-end delay, hence, this does decrease this metric on the overall. On the other hand, the power consumptions in the attacked scenarios are similar to that in the normal network scenario. Like the observation of authors in [85], we also saw that the 6LoWPAN RPL cannot eliminate this attack, most of the nodes in the coverage area of the attacker chooses it as the preferred parent during the simulation time.

Table 3.8. Performance comparisons between the normal and Sinkhole attack scenarios

Scenario	Global Delivery ratio	Average End-to-end delay (ms)	Total Overhead	Total collisions	Total Rank changes
Normal	1	531.27	371	98	111
Att at 9	0.55	419.98	506	135	147
Att at 8	0.57	620.46	487	165	143
Att at 10	0.64	414.85	499	131	153
Att at 6	1	524.55	392	103	121
Att at 22	1	692.05	410	167	137

3.5.1.3. Attack Behaviours

In any IDS, the monitoring node is normally not designed to monitor the sink. Therefore, it is difficult to verify whether a node advertises the rank of the sink is the legitimate sink or not, especially given that this node has managed to remain hidden by not doing any other attacking mechanisms. The only potential way to detect the Sinkhole attack is at the time the attacker initiates the Sinkhole attack. Before that time, this node should have a rank, which is not as good as the sink rank. Because the sink never changes the rank in a session, so when we observe that a node changes its rank to a value as good as the sink rank, the monitoring system should suspect this node of being under Sinkhole attack. This may be the only effective sign that helps to detect the Sinkhole attack.

3.5.2. Rank Attacks

The RPL routing rule stated that "rank strictly increases in the Downstream direction and strictly decreases in the Upstream direction" [18]. This rule is created to prevent the nodes from creating un-optimised path or loop path. Considering a scenario when the source - node 1 sends the packet to the destination - node N through intermediate nodes 2, 3, 4, …, n-1. Assume the rank of these N nodes are R_1, R_2, R_3, …, R_{n-1}, R_n consequently. The rank rule states that if node 1 sends packets upward to node N then the condition $R_1 \geq R_2$

$\geq R_3 \geq ... R_{n-1} \geq R_n$ must be satisfied; or if the route is downward then $R_1 \leq R_2 \leq R_3 \leq ... R_{n-1} \leq R_n$ must be satisfied. The senders and receivers along the route have the responsibility to check these conditions and inform any broken of this rule by setting the Rank-Error bit in the RPL Packet Information [18].

Once the rank rule is broken between a parent and a child node, the consequence can be (i) un-optimised path is created (ii) if the attack is initiated in the route discovery phase, some optimised paths may be disrupted, which mean they exist but will never be discovered, and (iii) a loop can be created without any detection. These consequences definitely downgrade the network performance in many important aspects, such as delivery rate, delay, and overhead.

Rank manipulation at first was not thought to be an attack mechanism. Only authors in [7] measure the healing procedure of RPL network operation after sudden changes in node rank. The experiment in this work is that after running for a pre-set time, a node increases its Rank to equal to the highest Rank value of its neighbours, and wait for a period of time to check the RPL healing process. The observed consequences include the appearance of loops between the node and its children and more control messages are generated, which indicate the unstable of the topology. Our research [8] [9] proposed that internal attackers can manipulate the rank rule to downgrade the network performance.

3.5.2.1. Attack Implementation

In [9], we presented in detail the four mechanisms that the attackers can use to twist the rank rule to break the optimal topology, based on the fact that the attackers can flip its best parent according to the rank and choose whether to update to other nodes. After the attack is triggered, the malicious node changes the way it processes the DIO messages from other neighbours so that it will choose a random node with a worse rank as the preferred parent during its operation. These four mechanisms are summarised in Table 3.9 as follows.

Table 3.9. Types of Rank attack studied

4 types of Rank attack	**Attackers choose whether making a permanent/non-permanent new parent, or flip the best parent over time**	
Attackers choose whether update/not update the new rank to the neighbours	RA type 1 (RA-I): having permanent new parent, updating the new rank info to its neighbours	RA type 2 (RA-II): having permanent new parent, not updating (hiding) the new rank info to its neighbours
	RA type 3 (RA-III): flip the new parent, updating the new rank info to its neighbours	RA type 4 (RA-IV): flip the new parent, not updating (hiding) the new rank info to its neighbours

The implementations of these four types are described through the pseudo codes as in Figure 3.11 (general manipulating mechanisms) and Table 3.10 (detail operation of each Rank attack type).

```
//Compromising the choice of parent by selecting the worst parent
1: best_parent(parent p1, parent p2) {
2:        if (node→attack_flag == TRUE) {
3:                return p1_metric > p2_metric ? p1 : p2;
4:                }
//p_metric: the cost of the route towards the sink
5: }

//DIO update_flag:only update DIO if the flag is disabled
1: If (node→DIO_update_attack_flag == FALSE) {
2: Update DIO; }

//3. Disabling DAO when having new illegitimate preferred parent
1: Select_preferred_parent(){
2: If (node→attack_flag == FALSE) {
3:        DAO_output;
4:        }
5: }
```

Figure 3.11. General Rank attack implementation

Table 3.10. Detail implementation of each RA type

//Type 1: Attack after t seconds 1: *node→att_flag* = FALSE 2: *node→DIO_update_att_flag* = FALSE 3: **If** (*time* > *t*) { 4: *node→att_flag* = TRUE; 5: }	*/*Type 2: Start attack after t seconds, disable the DIO update */* 1: *node→att_flag* = FALSE 2: *node→DIO_update_att_flag* = FALSE 3: **If** (*time* > *t*) { 4: *node→att_flag* = TRUE; 5: *node→DIO_update_att_flag* = TRUE 6: }

/*Type 3: Start attack after t seconds, then time is divided by multiple p-second periods, attack happen in only first half of each period*/
1: *node→att_flag* = FALSE
2: *node→DIO_update_att_flag* = FALSE
3: i**f** (*time* > t) {
4: *period = (time - 1)/p*;
5: *check = period – int(period);*
6: **if** (*check* < 0.5) {
7: *node→att_flag* = TRUE;
8: **else** { *node→att_flag* = FALSE;}
9: }}
	/*Type 4: Start attack after t seconds, then time is divided by multiple p-second periods, attack happen in only first half of each period. The rank info is not updated to the neighbours/*
	1: *node→att_flag* = FALSE
	2: *node→DIO_update_att_flag* = FALSE
	3: **if** (*time > t*) {
	4: *period = (time - 1)/p;*
	5: *check = period – int(period);*
	6: **if** (*check < 0.5*) {
	7: *node→att_flag* = TRUE;
	8: *node→DIO_update_att_flag* = TRUE;
	9: **else** { *node→att_flag* = FALSE;
	10: *node→DIO_update_att*_flag = FASLE;}
	11: }}

3.5.2.2. Impacts on Network Performance

Figure 3.12 to 3.15 show the comparison of performance metrics between the four Rank attack types and the normal scenarios, including the global delivery ratio, end to end delay, overhead, and number of collisions, respectively. As can be seen from the figures, RA-I and II have the least impact among the four types. RA-II has more effect on the end-to-end delay while RA-I has more impact on the delivery ratio. The nature of RA-I and II is very similar, the only difference is that RA-I allows other nodes around the malicious source to optimise the topology through updating the DIO messages while RA-II does not. As a result, the average end-to-end delay in RA-I is likely to be smaller than in RA-II. On the other hand, RA-I requires more additional control messages to maintain the optimised topology, so it may cause more packet collisions. This in turn reduces the delivery ratio to be less than for RA-II.

RA-IV on the other hand, has more probability of creating the greatest impact on network performance in both the delivery ratio and the end-to-end delay. RA-III of Rank attack has smaller impacts on global delivery ratio and average end-to-end delay compared with RA-IV, but significantly higher than RA-I and II. For these two types of attacks, the reason for having a higher delay than the first two types is that these two types create many changes in the topology by frequently changing the preferred parent of the malicious node. The nodes around the affected area also have to spend more time updating the route, which adds more delay to overall performance. The delivery ratio is decreased because more control overhead is generated, which leads to more packet collisions. RA-IV has a larger impact on end-to-end delay compared with RA-III because it does not

allow updating of the optimised topology, so there will be more non-optimised routes in the network.

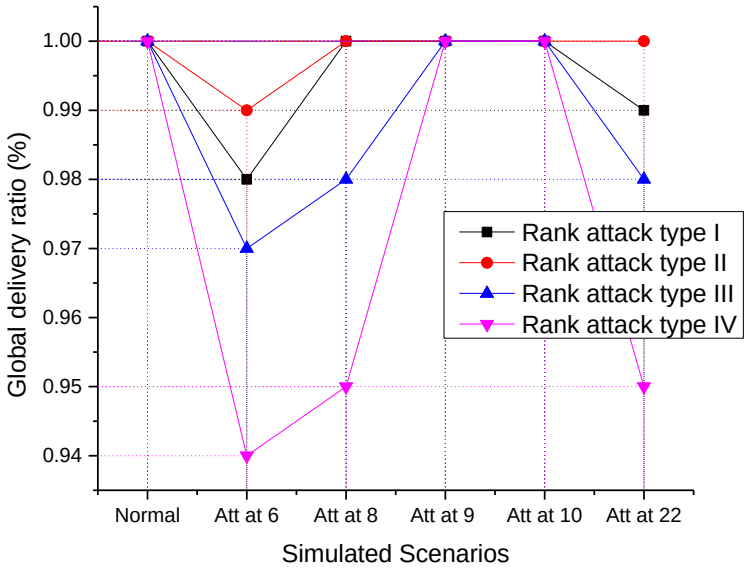


Figure 3.12. Comparison of the Global delivery ratio between the normal and four types of Rank attack scenarios

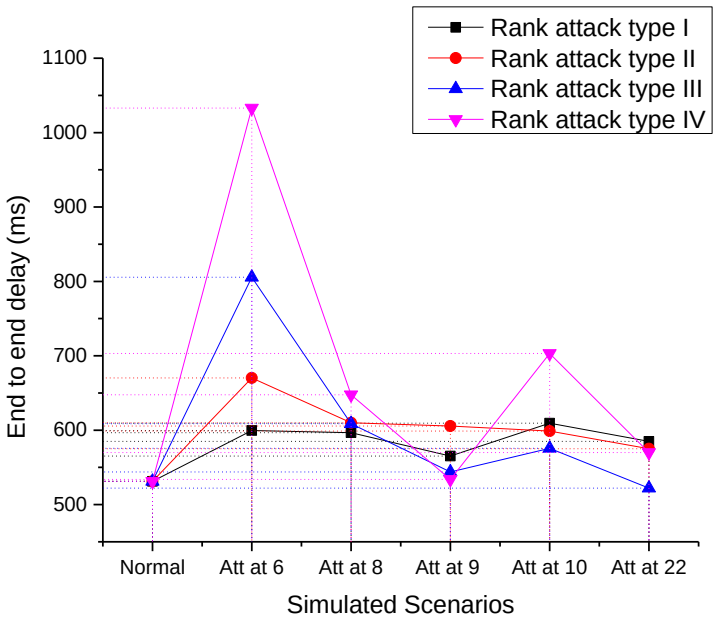


Figure 3.13. Comparison of the average End to end delay between the normal and four types of Rank attack scenarios

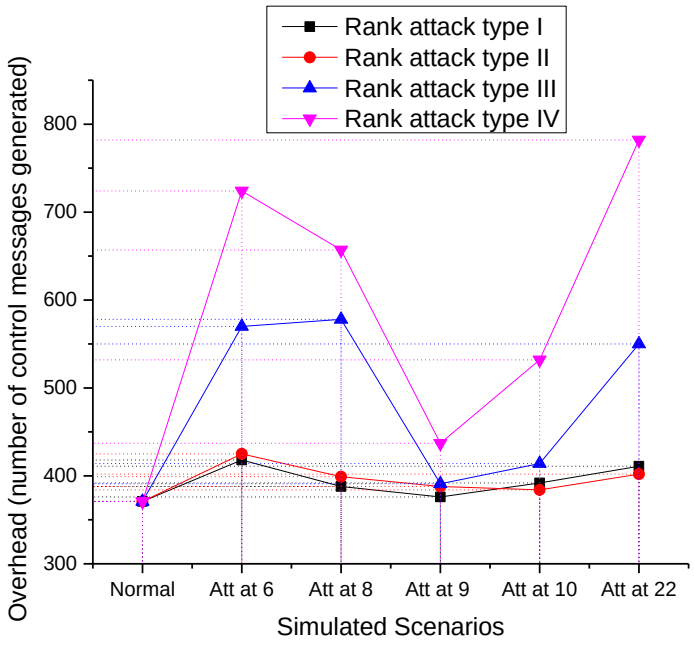


Figure 3.14. Comparison of the overhead between the normal and four types of Rank attack scenarios

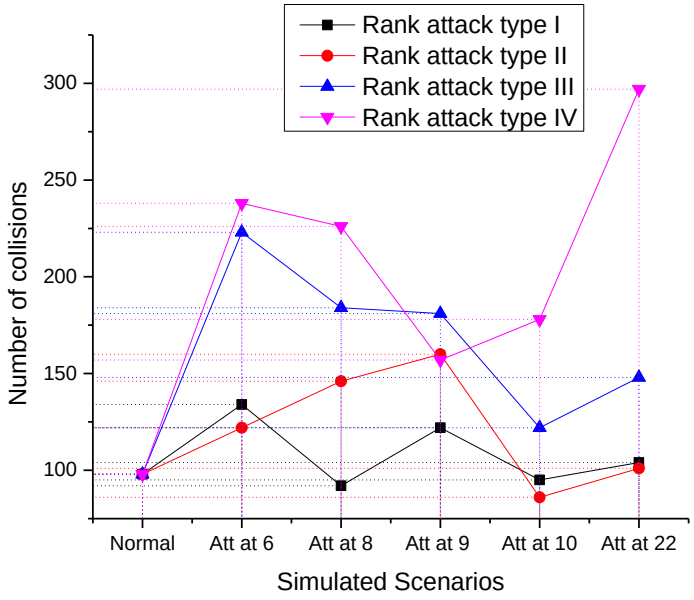


Figure 3.15. Comparison of the number of collisions between the normal and four types of Rank attack scenarios

3.5.2.3. Attack Behaviours

In cases of RA-I and II, the number of generated DIOs is larger compared with the normal performance, but that happened only for a short time after the attack. On the other hand, in cases of RA-III and IV, a significant number of DIOs were generated more constantly during the attack time. RA-III did not have many nodes that need to change the topology;

however, it still showed many generated DIOs. This suggests that the increase in the number of DIOs can have some other causes, for example, the change in the forwarding load of the neighbours around the malicious node makes those nodes distribute different routing information during this period. This, therefore, prevents the increase of DIO counter and DIO trickle time interval, and as a result, will make the node generate more DIOs than in the normal case.

In Rank attack, when the monitoring node looks at the performance of the nodes around, it can see the changes in the performance such as there are additional overhead and collisions created, however, it will not know the real reasons behind these anomalies. Therefore, it is necessary that the monitoring node specify and check the rank relation between the child and parent nodes frequently to detect this kind of attack.

3.5.3. Local Repair Attack

During the network operation, the link between a parent and a child node may not remain the same quality as it was in the route establishment phase. Hence, the maintenance phase is important for ensuring the quality of the link over time. The Local Repair mechanism is designed for RPL to re-establish a new optimal route between two nodes once the current link between them is broken. This is essential to maintain the optimal topology; however, the adversary can manipulate this mechanism to initiate more frequent Local Repair, which can lead to a frequent change in the network topology. We call such an attack the Local Repair attack [8, 10, 83].

3.5.3.1. Attack Implementation

A node in RPL can start the local repair progress in two ways. The first way is the poisoning mechanism by changing its rank to infinitive and broadcast this rank to all of its neighbours. Those neighbours once receiving and updating the rank information of that node will need to find a new parent towards the root. The second way to do local repair is to change DODAG ID value of the node. This metric is unique to each DODAG and show which LoWPAN the node belongs to. A node changes its DODAG ID means that it lefts that DODAG and now belongs to a new DODAG neighbour. As a result, all of its child nodes need to do a local repair to find for a new preferred parent. The main difference between these two types is that in the former case, the child node can still choose the adversary as a parent, while in the latter case, the adverse node will be

considered as not belong to the current DODAG, hence, may not be chosen as a parent node anymore.

Our implementation of the Local Repair attack in this thesis is similar to our implementation in [10], in which a malicious node starts broadcasting the local repair messages periodically though there is no problem with the link quality around the node. We consider the attack within a single DODAG only, so we skip the implementation for the DODAG type. The setting is that after the first 60 seconds of the network operation, the adverse node will start the local repair process every 30 times.

3.5.3.2. Impacts on Network Performance

Table 3.11 shows the simulation results of the Local repair attack to compare with the normal network performance scenario. As can be seen from the table, many performance metrics were affected significantly, including the average end-to-end delay (up to 33.7% slower), the total collisions (up to 3 times larger), and the total rank change (up to 140% higher). The reason is that every time Local Repair attack is generated, the route establishing and maintenance functions are invoked not only in the compromised node, but also in its child nodes because of the detachment of the parent. The route establishment and maintenance require related nodes to recalculate their rank from start; therefore, these procedures will involvea lot of control messages traffic and rank changes. On the other hand, the global delivery rate remains the same, which indicates that the route update only delays the message sending but not dropping them.

Table 3.11. Performance comparisons between the normal and Local repair attack scenarios

Scenario	Global Delivery ratio	Average End-to-end delay (ms)	Total Overhead	Total collisions	Total Rank changes
Normal	1	531.27	371	98	111
Att at 6	1	532.39	813	141	101
Att at 8	1	614.69	837	192	150
Att at 9	1	630.57	790	333	161
Att at 10	1	707.56	831	195	135
Att at 22	1	710.8	848	204	112

3.5.3.3. Attacker Behaviours

From the view of the monitoring node, the monitored node will initiate the Local Repair mechanism frequently. In RPL, the node is supposed to do the local repair only if the links towards its parent list are all broken. The Local Repair adverse initiates the Local Repair not because of this reason; however, there is no way that the monitoring node can

verify that. Therefore, it cannot justify whether that the monitored node is benign. The only way to differentiate between the normal and attacker node is to add statistics in monitoring the frequency of a node doing Local Repair mechanism. The underlying symptom for detecting the attacker is that the normal node does not have many Local Repair initiating more than a threshold number in a periodic of time.

3.5.4. The DIS Attack

One of the most important requirements for designing RPL is that the routing overhead needs to be limited as much as possible, to preserve the resource and lengthen the network lifetime. On the other hand, attackers are always interested in making the nodes waste more resources by creating overhead. Attackers presented in the previous sections have the ability to increase the routing overhead; however, it is not direct but only as the consequences of other mechanisms. In this section, we introduce the DIS attack as a direct way to keep the overhead of the neighbours to be always in high level [10].

For this kind of overhead attack, we focus on the trickle algorithm (see Section 2.2.2.2 for more detail), which is specifically designed for RPL to limit the overhead created. The trickle value is increased through time if there is no significant change on the routing topology, for example, new node joining the network, or Local Repair and so on. A high trickle value indicates that a smaller amount of control messages will be generated. The DIS attackers try to keep this value in its neighbours to be minimum by frequently sending the DIS. The neighbours once receive the DIS will understand that the topology is unstable, and sending control messages more frequent to keep update the information. Consequently, the minimum trickle time may be set in a wide range of area in the network, which create significantly overhead. If this continues, the overheads are created more and more, the node becomes exhausted faster and network performance will be downgraded.

3.5.4.1. Attack Implementation

Our implementation is similar to our work in [10]. The main purpose of the attack is to increase more DIOs through manipulating the use of DIS. The DIS messages can be sent two ways, which will lead to a different response from the receivers. The first way is to broadcast DIS, the receivers upon receive will have to reset the DIO timer as they realise that there is something unstable with the topology around. The second way is to unicast this DIS message to all nodes in the neighbour list, the receivers upon receive will unicast DIO message to the sender. Both of these ways add more control overhead on the

network. The differences of these two types of DIS attacks are that one type (DIS1) forces the receivers to decrease their DIO timer while the other (DIS2) forces the receivers to unicast their DIO messages to the compromised node. For simplicity, in this thesis, we only assess the behaviour of the DIS1. Detailed assessment of these two attacks can be found in [10].

3.5.4.2. Impacts on Network Performance

Table 3.12 shows the simulation results of the DIS type 1 attack to compare with the normal network performance scenario. As can be seen from the table, the global delivery ratio remains the same, which indicates that this attack does not lead to packet dropping. On the other hand, the routing overhead creates by decreasing the trickle time make the total number of generated control messages increase significantly, from 163% to 194%. This leads to the following changes. First, the collisions are increased up to 91% because of more communication made caused by new routing information was updated. Second, ranks are changed more frequently up to 81% due to the unstable topology. Third, the average end-to-end delay is slightly increased up to 26% due to the retransmission created by the collisions together with the waiting time for processing the control messages.

Table 3.12. Performance comparisons between the normal and the DIS type 1 attack scenarios

Scenario	Global Delivery ratio	Average End-to-end delay (ms)	Total Overhead	Total collisions	Total Rank changes
Normal	1	531.27	371	98	111
Att at 8	1	572.12	605	110	137
Att at 22	1	608.97	674	116	128
Att at 10	1	614.44	612	140	180
Att at 6	1	650.78	722	151	151
Att at 9	1	671.96	642	187	202

Figure 3.16 shows the consumption of energy regarding the computation and communication in each of the simulated scenarios. As can be seen from the figure, the computational waste increases significantly from 48% (when attacker at node 8) to 76% (when attacker at node 9), while the communicational waste increases from 9% (when attacker at node 8) to 14% (when attacker at node 9). The energy waste was mainly caused by the processing and communicating of the additional control overhead. A further checking through time showing that unlike the normal scenario, where the overhead is decreased through time, the waste in the attacked scenarios almost remains the same. This indicates that the DIS attack can significantly shorten the network lifetime.

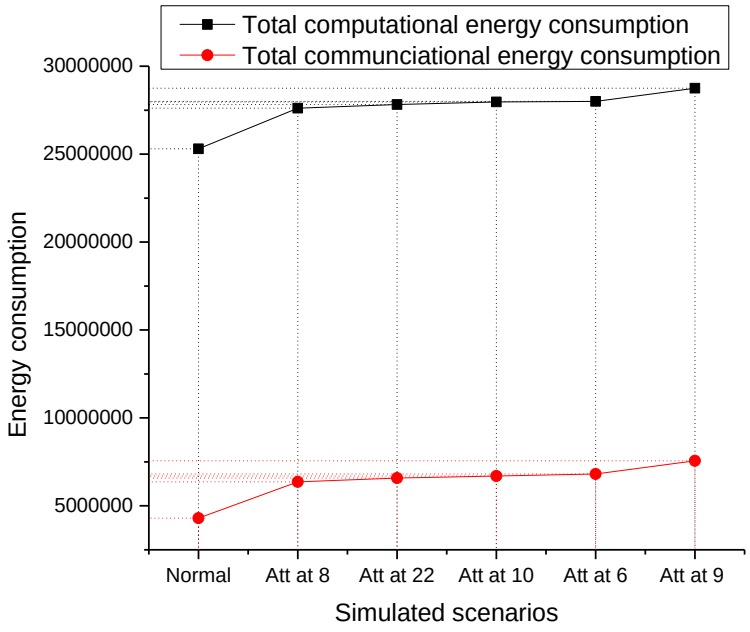


Figure 3.16. Comparison of the computational energy consumption between the normal and DIS type 1-attack scenarios

3.5.4.3. Attack Behaviours

The only difference between the DIS attacker and the normal node is that it broadcasts the DIS messages frequently. The monitoring node cannot check the reason behind this action, therefore, it cannot decide whether the node is benign or not. Nodes around the adverse node have many anomalous activities such as sending control messages more often, updating rank many times more, or having more collisions. However, these activities are caused by the sending of DIS messages from the adverse rather than the direct manipulation in the nodes themselves. Therefore, to detect this kind of attack, the DIS processing needs to be profiled to form the legitimate pattern.

3.6. Chapter Summary

This chapter has discussed several typical internal threats towards 6LoWPAN, categorised into two types: the performance-type and the topology-type. We summarised the impacts of the attacks and the key behaviours of the attackers in Table 3.13 below. The impacts of each specific attack on the network performance are also illustrated by the impact map in Figure 3.17, represented through the three dimensions, including the End to end delay, the delivery ratio, and the control overhead generated. The sensitive metrics to specific type of attacks are also summarised in Table 3.14.

Table 3.13. Summary of prominent internal attacks towards 6LoWPAN

Type of attacks	Attack impact	Key behaviours of attackers
Black Hole/Grey Hole attack	**Black Hole:** change the global delivery ratio significantly, especially when attacker sitting near the sink **Grey Hole:** downgrade global delivery ratio	Having low forwarding rate given a significant traffic goes through
Delaying attack	Adding global end to end delay, increase dropping rate, gaining more overhead, collisions, topology changes, and consume significant more energy	Abnormal increase in forwarding delay; update the rank metric more often; sending more DIO
Jamming attack	Increase the global end-to-end delay, create more collisions on the network, which leads to consume more energy	Abnormal increase in collisions around the attackers
Hello Flood attack	Increase the global end-to-end delay, total overhead, total collisions and total topology changes	Abnormal high signal strength; advertise good routing value, but little traffic goes through
Sinkhole attack	Increase the global end-to-end delay, packet dropping rate, and overhead	Time when a node changing its rank from a low value to the sink rank
Rank attack	Adding delay, reduce delivery rate, adding control overhead	Abnormal in choosing preferred parent (break the rank rule), frequent change in rank
Local repair	Adding delay, control overhead, and collisions	The frequency of local repair is abnormal
DIS attack	Increase overhead, rank changes and collisions	The abnormal in sending DIS

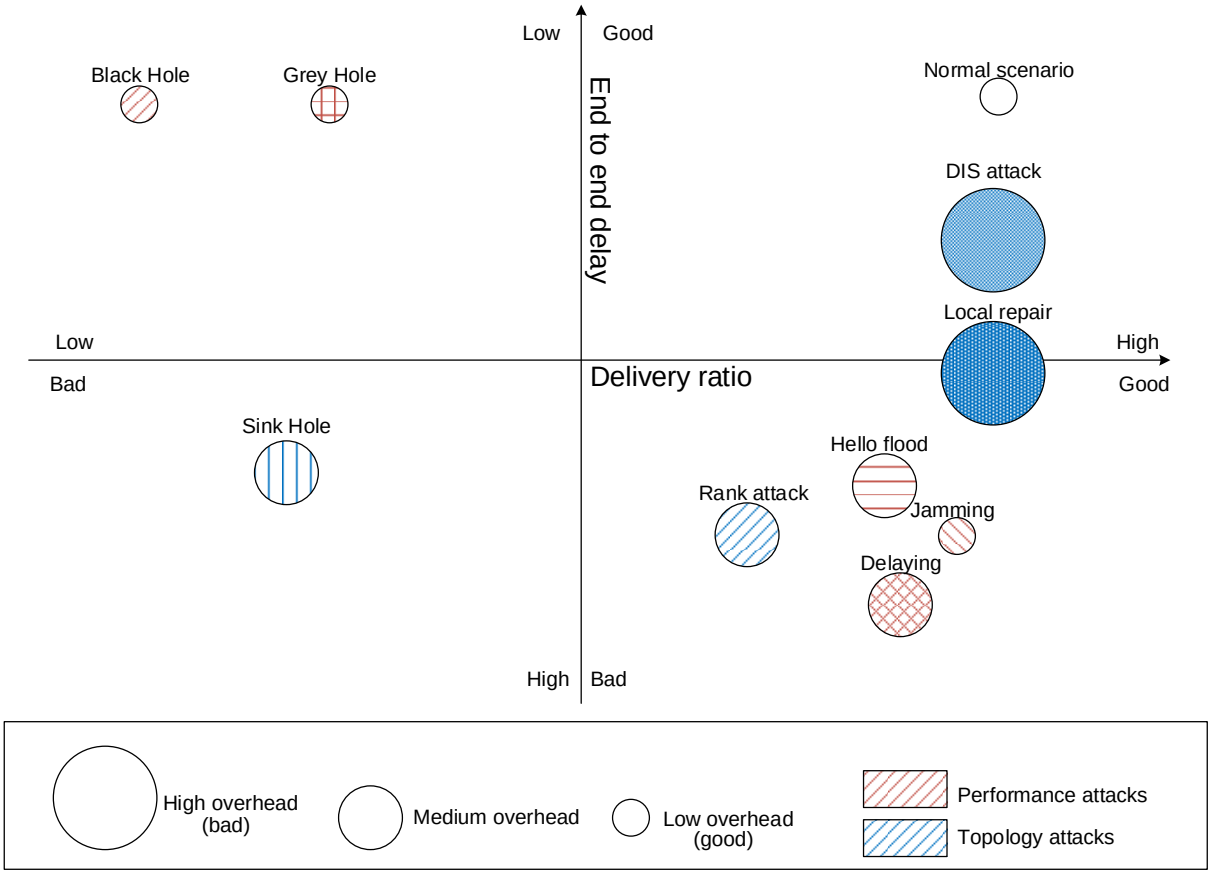


Figure 3.17. Impact map of internal threats toward 6LoWPAN RPL performance (End to End delay, Delivery ratio, and Overhead).
Red colour indicates the direct performance attacks, Blue colour indicates the topology attacks, and the size of the circle indicates the routing overhead created.

Table 3.14. Summary of sensitive metrics to each attacks

Type of attacks/Metrics	1	2	3	4	5	6	7	8
Black Hole								
Grey Hole								
Delaying								
Jamming								
Hello Flood								
Sink Hole								
Rank								
Local Repair								
DIS								

Metrics list: 1) Forwarding Rate 2) Forwarding Delay 3) Power Consumption 4) Packet collision 5) RSSI change 6) RSSI value 7) Number of Rank changes 8) Number of generated control messages. The green colour indicates that the metric is sensitive to the Bayesian model

Overall, we can see that the performance-type attacks aim at the traffic, channel, or neighbour operation around the adverse, while the topology attacks focus more on the protocol process. Although both of these internal threats aim at downgrading the network performance, they are different in nature. As a result, different IDS should be developed to detect these two types. IDS which focuses on the local performance of the nodes is best to detect the first type of attacks, while IDS which concentrates on a proper protocol

specification is best to detect the second type. Our system will therefore be designed to deal with these two types of internal threats separately. In the next chapter, we will present the BN method for detecting the performance-type attacks by collecting and judging node behaviours statistically from the collected data.

CHAPTER 4. A BAYESIAN BASED IDS FOR DETECTING THE PERFORMANCE THREATS

4.1. Introduction

The last chapter studied the impacts of the internal attacks towards the network performance and the specific characteristics of the malicious source. This chapter will deal with the first type of internal attackers in 6LoWPAN. In detail, knowledge from the last chapter will be used to develop an IDS for detecting the typical performance-type threats.

Performance attackers usually have significant anomalous behaviours such as packet dropping, delaying, and collisions with neighbours and so on. As a result, anomaly-based IDS is often applied in detecting such internal threats. Proposed solutions in anomaly-based IDS normally focus on a single metric to deal with particular attack, for example, the dropping rate metric is usually used for detecting the Black Hole or selective forwarding attack. Even in a system which is claimed to protect the network from multiple internal attackers like in [75], the author combine the list of key metrics for the common attacks, yet use them separately to deal with each attack. The use of a single performance metric as symptom to judge an attack may lead to limited views of the node behaviours, which may mislead the judgment. For example, we consider the use of only the dropping rate metric to justify whether a node initiates a selective forwarding attack in [75]. According to the author, a node with a high dropping rate is likely the selective forwarding attacker. However, a legitimate node which has a lot of forwarding load while having low energy left may also drop a lot of packets. Regarding the drop rate metric only, the behaviours of this node are similar to the behaviours of the selective forwarding attackers. In such a case, the use of this single metric may lead to a false judgment because the deviation between the legitimate and the malicious nodes' behaviours is small.

Chapter 3 has shown that for threats that aim at downgrading the network performance directly, there are local metrics of a node, which are sensitive when this node has abnormal behaviours. Therefore, if we can observe the suspicious change of such metrics, we will be aware of the occurrence of an internal attacker. The nature of this problem makes it suitable to be solved by Bayesian Network model, where these metrics can be considered as the evidences to judge a behaviour. The BN model can provide a wider and

deeper view in terms of considering multiple evidences, which will lead to detection that is more accurate. Other advantages of BN model can be found more in Section 2.5.2.4. However, to the best of our knowledge, at the time of writing this thesis, there is no proposed BN solution in detecting 6LoWPAN internal threats. The reasons are that Bayesian method requires quantifying computation load and memory storage, which may lessen the node lifetime when implementing in 6LoWPAN. Moreover, in order to construct the Bayesian model, a significant amount of data regarding node performance need to be collected and processed to train the model, which is quantifying to do in 6LoWPAN.

In this chapter, we will develop a Bayesian model, which takes into accounts the sensitive metrics that relate to the performance-type, and their interrelations. We first present a designed architecture that combines 6LoWPAN with other technologies likeWi-Fi, 4G, 3G or GSM, cloud computing, and multiple interfaces in sensor devices, to enable the use of BN in 6LoWPAN. With our proposed architecture, we believe that the IDS is capable of deploying more robust analysis techniques due to the extensive storage and computation capacity. Given the available list of the sensitive metrics and the large amount of data, which can be collected, the integrated BN can provide high accuracy by integrating the insightful expert understanding of the relations between the evidences in assessing a phenomenon. Besides, it can predict even with missing observed data, which gives flexibility for the detection. We describe the construction of the BN structures (nodes, edges and parameter) from understandings of the performance threats and training with the extensive simulation. After the construction, BN model can give numerical assessment for behaviours of the monitored nodes. Such model is then integrated with the monitoring nodes. Next time, given the real time data collected from the monitoring nodes, the BN module can give the judgement for every observed behaviour. Lastly, we check the effectiveness of the model through assessing its detection ability towards the typical attacks in Chapter 3, including the Black Hole, Grey Hole, Delay, Jamming, and Collision attack.

4.2. Enabling the Use of Bayesian Technique in 6LoWPAN

The main issues when applying Bayesian technique are the requirements of processing and storing capability in the monitoring nodes given the high computation workload and large amount of IDS data. This issue can be solved by delegating such work for an IDS server. In the traditional IDS for WSN, such IDS server is the sink, which is normally a

computer and has strong capability in processing and storing data compared with other sensor nodes. Nevertheless, placing IDS in the sink has several limitations. First, the communicational overhead created by sending the IDS data to the sink is significant and will increase quickly when network scale increase, which limit the scalability of the method. Moreover, a common understanding of IDS is that the more data that the system can collect, both in the number of monitoring metrics and the length of the monitoring duration, the more accurate it is. However, data collection involve cost and management challenges if recorded in a long time. Given the large number of nodes in 6LoWPAN, even recording a single metric in a certain long period can create storage and process issue for the sink as a single computer. Recently, the advance in Big Data research can help to solve this problem [91].

To eliminate the communicational overhead, the IDS data should be ideally sent through a different channel, dealing with the extensive IDS workload, the IDS server should have much higher processing and storing capability than the sink. The former goal is solvable by providing an additional interface in the monitoring nodes to allow them to send IDS data directly to the IDS server. Research on allowing sensor nodes to have multiple interfaces has been started since mid-2000, for example, the work in [92, 93]. Recently, developments in technology enable sensor devices to have interfaces, which can send data with high speed, long distance, and low power consumption through standards like Wi-Fi, 4G, 3G or GSM. The IDS data therefore can be effectively sent to the IDS server using such technology. The use of such backbone to transmit the monitoring information to the server is feasible, given such technology is popular and having low price on implementation. On the other hand, the latter goal can be solved by employing technology like Cloud Computing [94]. With Cloud Computing, the storage and computation work will be shared by other computers, so the capability is extended much more, while the cost for implementing is modest. The extension of this design will allow to apply not only Bayesian technique but also many other complex detection algorithms with high accuracy to detect the internal threats in 6LoWPAN.

Figure 4.1 below illustrates the idea of using additional interfaces and cloud computing to enable the use of BN in 6LoWPAN IDS. In this design, the monitoring nodes (brown nodes) use their 802.15.4 interface to communicate with other nodes like normal, to fulfill the 6LoWPAN role. On the other hand, they have an additional interface, which allow them to send data directly to the IDS server through the base station. IDS data of the monitored nodes will be collected and preprocessed in the monitoring nodes before

sending. At first, the IDS server can use the collected data to train the BN module. After the training phase, it can use the constructed BN to assess the monitored nodes' behaviours. The processing and storing capability of the IDS server can be expanded significantly when it runs on cloud-computing infrastructure to share the workload with other computing resources in the cloud.

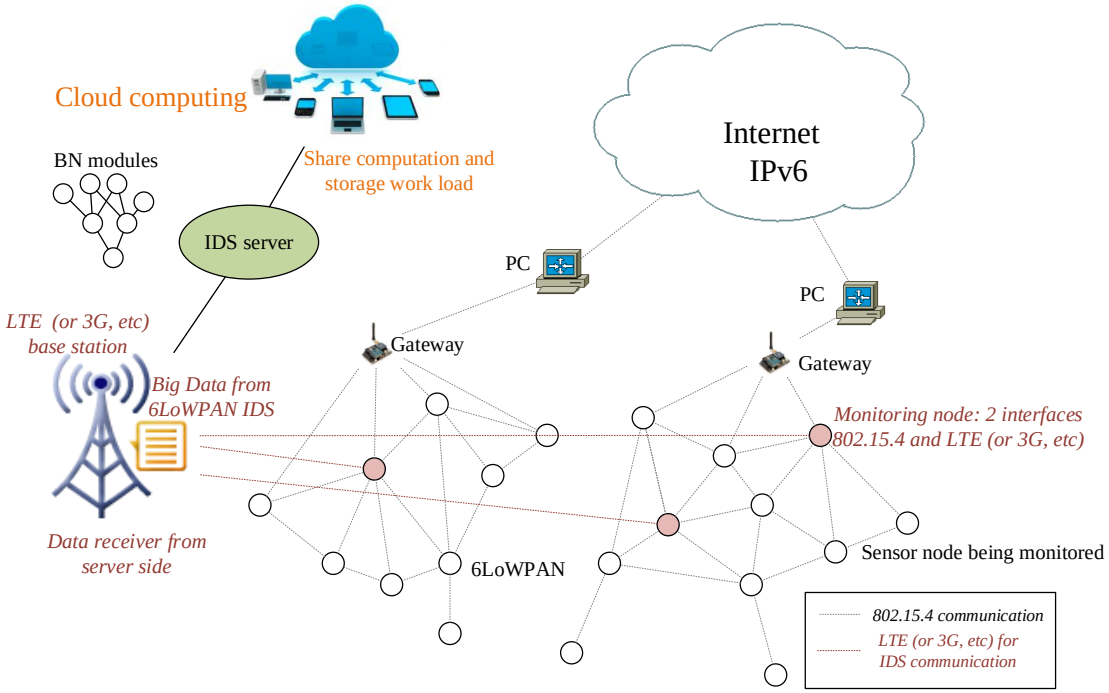


Figure 4.1. Design for enabling Bayesian technique in 6LoWPAN IDS

4.3. BN Constructing Process

Section 4.2 has pointed out the design that enables the use of Bayesian IDS in 6LoWPAN. In this section, we will present our general step-by-step procedure to construct a BN to judge behaviours based on observed evidences. This procedure is formed through our extensive study of the Bayesian literature. The background of BN can be found in Section 2.5.2.4. Here we only present the four main steps to construct the BN, which are illustrated in Figure 4.2 below:

- **Nodes forming and data set acquiring:** the main objective of the BN is to answer a query about a phenomenon based on the observations of different causes that relate to the query. The first step of constructing the BN is therefore to identify the metrics that represented these causes and acquire the data set of these metrics through time. Such metrics will later be the Bayesian nodes.
- **Structural learning:** the structure of a BN reflects the relations between their nodes. Such relations can be learned statistically based on the data set acquired in

the first step. These relations will later be the edges between the Bayesian nodes. The direction of the edge represents the cause and effect relation, i. e. the cause is the parent node (tail of edge arrow), while the effect is the child node (head of the edge arrow).

- **Parametric learning:** this step will calculate the parameters such as Bayesian nodes' probability and CPT in order to enable the BN to do the querying.
- **Implementing the constructed Bayesian model:** This step presents the way to use the model for judging the node behaviours based on the recorded data.

The following sections will explain how the steps are processed.

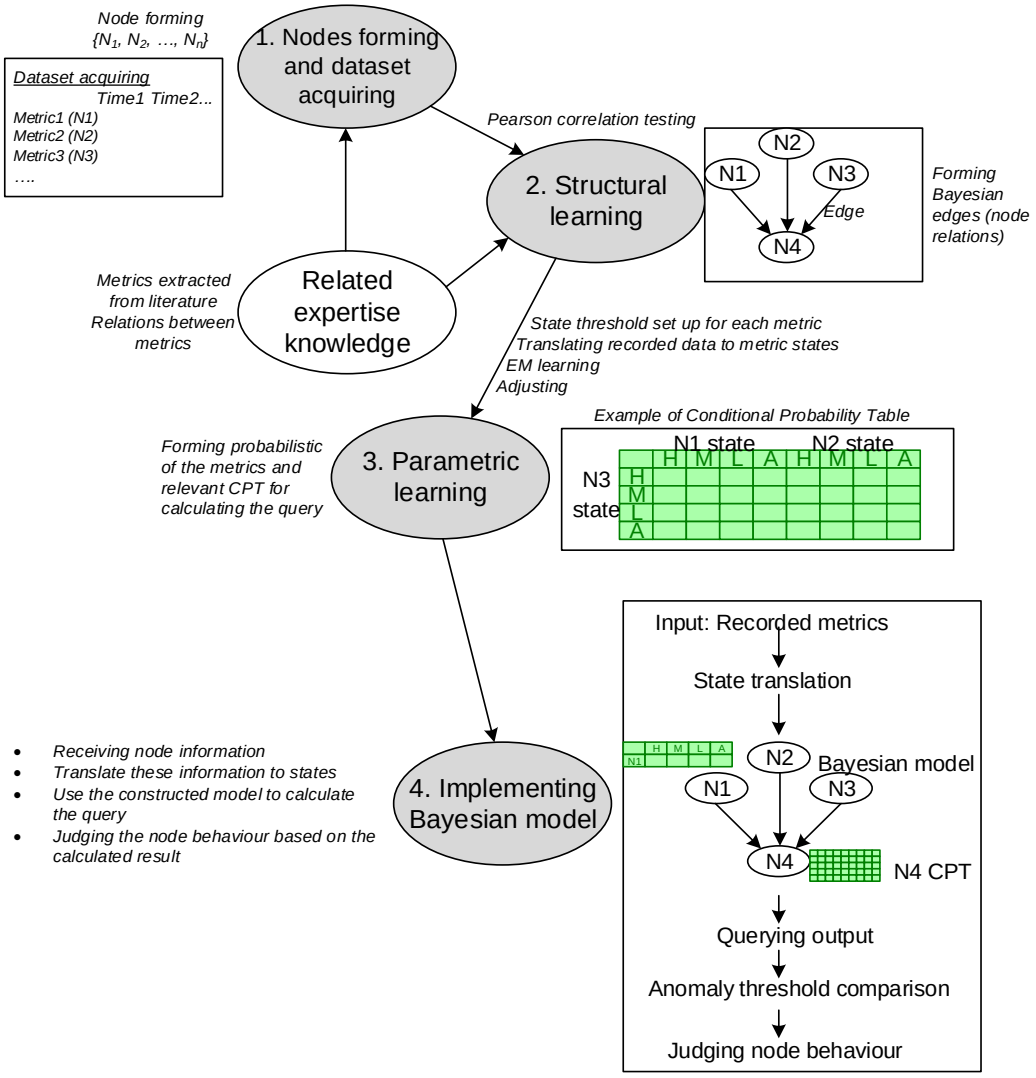


Figure 4.2. Steps to construct a BN

4.3.1. Nodes Forming and Data Set Acquiring

In order to answer the query about an effect, first, all the relevant causes that lead to this effect need to be identified. The chosen causes reflect the understanding of the Bayesian

developer about the effect, so they can come from both the literature and the experiments. In our case, the causes are all the related metrics that have sensitive reactions to the change of the attackers' behaviours. Such metrics have been derived from the literature as well as assessed through simulations as shown in Chapter 3. After the identifying process, we will have a set of metrics $N = \{N_1, N_2, \ldots, N_n\}$, which represented as the nodes of the BN. For each metric, we acquire its data set by sampling through a window time with fixed length, in order to see both the normal and the abnormal changes in the node's behaviours. This data will serve as the training data to form the relations between the metrics.

4.3.2. Structural Learning

This phase will build the structure of the Bayesian model through justifying the relations between the nodes. The two common approaches to form the relations are the supervised and unsupervised learning. The former integrates the expert knowledge related to the model to decide the relations, while the latter obtains the relations based on some probabilistic testing models through on the acquired training data. The supervised learning approach can provide the expert understanding of the querying model; however, it does not reflect the training data, which may vary depending on different scenarios.

Given the acquired data in the time series form from Section 4.3.1 and the knowledge of the effect from Chapter 3, we will be able to apply both of the supervised and unsupervised approaches in learning the Bayesian structure. In detail, the relation between a pair of node N_i and N_j in the BN can be assessed through a probabilistic method to test the correlation between the data set of these two nodes. If the test showed a statistically significant correlation, the relation will be represented as a connected edge between these two nodes, otherwise, there is no edge to be assigned. The direction of the edge will come from further reasoning of the cause-effect relations between these particular nodes. At the end of this phase, a set of edges $E = \{E_1, E_2, \ldots, E_m\}$ will be performed which represent all the learned relations of the Bayesian nodes. Such relations reflect both the acquired data in Section 4.3.1 and the knowledge of the effect through the literature.

4.3.3. Parametric Learning

This phase will set the probabilistic parameters to enable the Bayesian model to calculate the assessment query. In detail, the probability of each Bayesian child and the CPT in the parents need to be identified. As the BN requires the discrete input in each node, we first

need to specify the method to transform a numerical input of a metric to a state input. This can be done through setting the upper and lower numerical thresholds for each state of a metric. The procedure to set these thresholds for 6LoWPAN Bayesian will be presented in more detail in Section 4.4.3.1. Given the BN structure from Section 4.3.2 and the data set in the state form, we can use specific BN software to learn the BN parameters. For example, a tool like SamIam [95] can be used to implement the parametric learning algorithm (i. e. Expectation-maximisation learning (EM)) to generate the BN probabilistic values. Moreover, the tool also helps to further increases the accuracy of the model by providing the sensitivity analysis function, in which users can apply some constraints to the BN model to adjust the inaccurate answers given by the model.

With the first three phases, we can construct a BN, which is able to give the answer to a query regarding the phenomenon. In case of 6LoWPAN node behaviour monitoring, we can feed the numerical input data of the metrics to this model. The model will then give a numerical output, which reflects the behaviour assessment.

4.3.4. Implementing the Constructed Bayesian Model

This phase first shows how the constructed Bayesian model be applied to the monitoring nodes. In detail, the monitoring nodes will collect the numerical data of each metric from the monitored nodes periodically as the input. The Bayesian model will process such data through three modules as follows:

i. The input module, which takes the numerical-state converting reference of each metric to transform the numerical input to the state input;
ii. The calculation module, which takes the input as the state form to feed in the constructed BN, and give a numerical output regarding the probability of each states of the query;
iii. The decision module, which transforms the numerical outputs to the final assessment of this behaviour.

After processing through these three modules, the output of a sample data from a monitored node can be either Normal or Abnormal depending on the detection threshold. During network operation, the sampling data regarding the relevant metrics of a monitored node will be sent to the monitoring node periodically. The integrated BN

model in the monitoring node will give the decision for each of the monitored node's behaviours based on the data it collected.

4.4. Proposed Solution

In this section, we apply the steps presented in the last section to construct a Bayesian IDS for 6LoWPAN. The data flow of the whole process is illustrated in Figure 4.3 below. As can be seen from the figure, in order to train the BN, the monitoring nodes need to send relevant data sets of their monitored nodes to the IDS server. In this chapter, we will reflect this work through processing the Contiki Cooja simulation trace files as a proof-of-concept. We first collect the trace files from Contiki Cooja simulations. To extract the relevant metrics, before the simulation, we need to mark the relevant Contiki code to output in the trace file the part that contains the corresponding information. For example, in order to derive the end-to-end delay metric, we need to mark the times when a packet was sent and received (the packet is characterised by its sender's ID and the sequence number, and the end-to-end delay is the difference between the sent and received time). The list of metrics, which need to be derived, are justified in Section 4.4.1. The output of Step 1 is a table contain numerical data of metrics as shown in Figure 4.3. These numerical data will be used to test the correlations between the metrics to build the structure of the Bayesian model in Step 2. Before training this model, we need to set up a numerical-state converting reference through the approach, which will be presented in Section 4.4.3. Such reference will later be used to convert the numerical data from the output of Step 1 to the state form. The converted state data will be used to train the system to generate the probability of the child nodes and the CPT in each BN parent node in Step 3. Finally, in Step 4, for testing the module in each scenario, we implement the Bayesian module on the IDS server and run the simulations to check the effectiveness of the IDS. The details of these processes are described in the next sections.

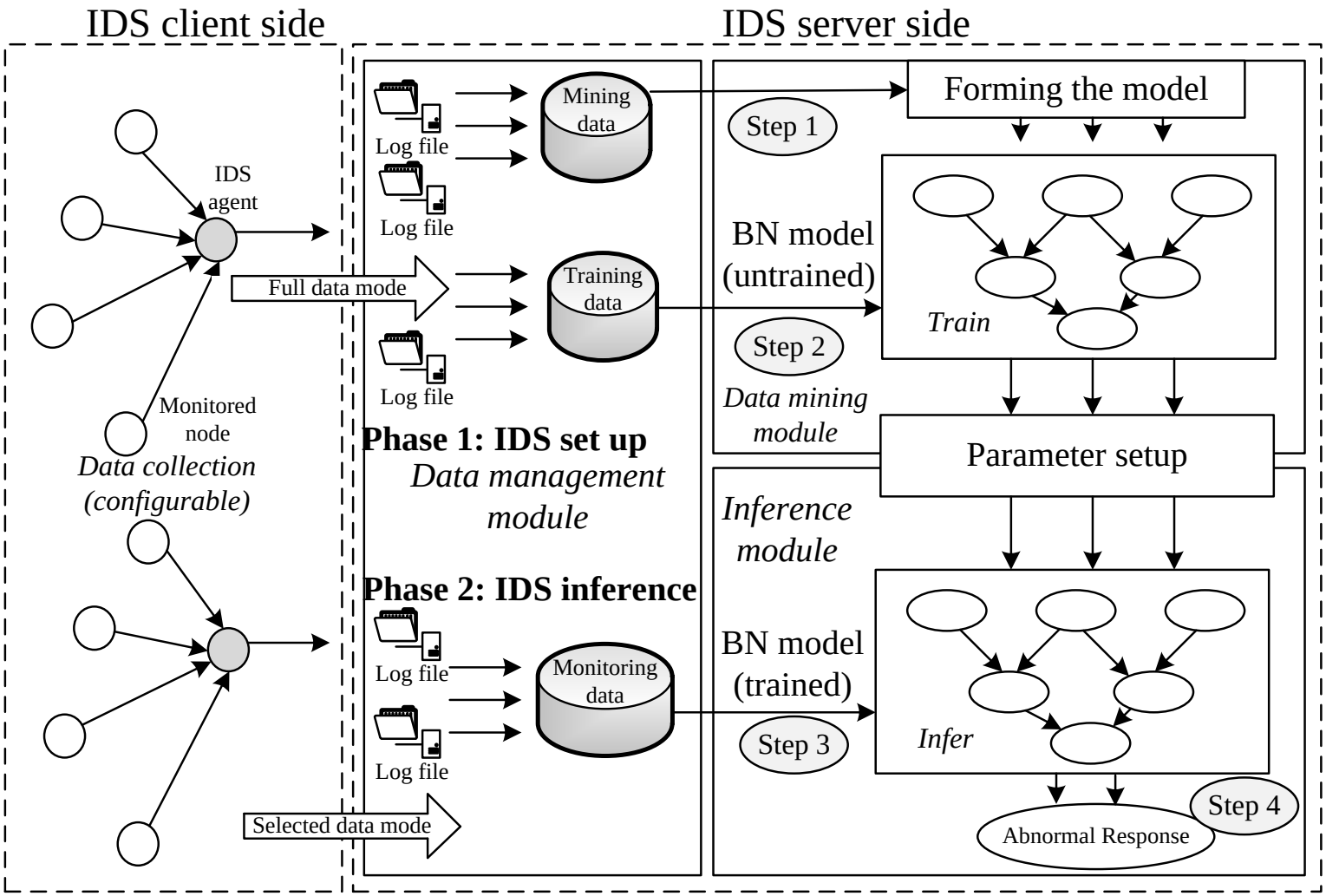


Figure 4.3. Detailed steps to construct a BN

4.4.1. Nodes Forming and Data Set Acquiring

4.4.1.1. Nodes Forming

As discussed in Chapter 3, the internal attackers can downgrade the network performance through the three main ways:

1. To manipulate the traffic that goes through it
2. To manipulate the communication channel
3. To disrupt neighbours operations.

We will focus only on the local metrics that are sensitive to these three types of behaviours.

The selected metrics to be considered in this chapter are as follows.

- **Forwarding rate:** represents the number of packets that are forwarded out of the number of packets that need to be forwarded. This metric is sensitive when the monitored nodes have the dropping behaviours.
- **Forwarding delay:** represents the average time between receiving a packet, and forwarding this packet to the next neighbour. This metric is sensitive when the monitored nodes have the adding delay behaviours.
- **Power consumption:** represents the computational and communicational consumption of a node. This metric is sensitive when the monitored nodes create channel interruption. For example, if a node has high power consumption but it does not receive or forward many packets, this node is suspected of using the power for jamming or interrupting the channel, which affect other nodes' performance.
- **Traffic load:** represents the number of data packets that were sent to this node for forwarding purpose. This metric is mainly used in accordance with other metrics to further cross check the legitimate behaviours. For example, nodes, which have high traffic load, will not be able to have high performance like nodes with low traffic load because they have more packets to process. Without considering this metric, the IDS may not understand the correct reason behind a downgrade of node performance; hence, the detection will have lower accuracy.

- **Packet collision ratio:** represents the state of the communication channel. This metric is sensitive when the channel is under attack.
- **Signal strength:** this metric is obtained from the RSSI values of the monitored nodes that are measured at the monitoring node. Our Bayesian model differentiates RSSI into two metrics, which are the *RSSI value* and *RSSI change*. The *RSSI value* metric will be out of a normal range when the attacker increases its power to send the control messages further than its coverage range, to create the confusions of the neighbour ID. On the other hand, the *RSSI change* metric records the number of abnormal changes that the monitored nodes have. The RSSI can be lowered through time, but it cannot change too much in a period. Therefore, if we observe significant changes in the RSSI of a node, we suspect that this node may have fake/replicated ID issue.

A summary of the features and their use is given in Table 4.1 below.

Table 4.1. Features used for the BN

Metrics	Usage
Forwarding rate	Sensitive to the dropping behaviours
Forwarding delay	Sensitive to the delaying behaviours
Power consumption rate	Sensitive to the channel interruption behaviours.
Traffic load	Used in cross-check with other metrics
Packet collision ratio	Sensitive to abnormal channel behaviours
RSSI value	Sensitive to the neighbour attacks, which create RSSI out of normal range
RSSI changes	Sensitive to the fake/replicated ID attacks, in which one node has too different RSSI value in small periods.

4.4.1.2. Data Set Acquiring

We acquire the dataset of the justified metrics from the last section through simulation trace files. We run two types of simulations with the same network topology set up, the normal network condition and the attacked condition. For the attacked condition, we initiate the malicious code of each of the four performance-type internal threats in several positions of the network, similar to the set up in Chapter 3. For each simulation, we record relevant metrics in each node through sampling in a fixed window time. In order to do so, we add some relevant "marking" in the normal Contiki OS code, so as later these "markings" will help to extract the metrics from the log files. For example, to extract the "Forward load" metric, we first mark the Contiki OS code in a way that can output the time when a node receives data packets that required further forwarding, and the time that this node actually forwards this packet. We count the number of packets that were

received and forwarded to that node in a period through the window time, and record them as the data samples for the "Forward load" metric of the node. We also marked the state of the behaviour in each time, in which all the behaviours from the normal nodes are marked as FALSE, while all other behaviours from the malicious nodes after running the attacking code are marked as TRUE. These acquired data will be used for further BN training. Table 4.2 below show several examples of these acquired data.

Table 4.2. Example of dataset acquired from a node through window time

Metrics/time	**Period 1**	**Period 2**	**Period 3**	**Period 4**	….
Forwarding rate (%)	100%	100%	100%	85%	
Forwarding delay (ms)	257	364	854	1274	
Power consumption (%)	151051	217028	303834	236852	
Traffic load (no of packets)	2	3	4	3	
Packet collision ratio (%)	4	5	7	4	
RSSI change (%)	0	0	0	0	
RSSI value (dBm)	-54	-54	-54	-54	
ATTACK	**FALSE**	**FALSE**	**FALSE**	**TRUE**	

4.4.2. Structural Learning

Our ultimate purpose of constructing a BN model is to use it to answer the query that whether a node's behaviour is abnormal. This query is represented as the final node "IsAttack" as shown in Figure 4.4. In order to answer that query, we have used the seven metrics that are sensitive to the performance-type internal threats in the last section. Every behaviour of the monitored node will be assessed through these metrics. In case the nodes are normal, the collected data regarding these metrics will always be in a certain legitimate range. On the other hand, if an attacker initiates any performance-type attack, defender will be able to observe some abnormal patterns in the recorded data. Such anomaly will be the evidence that will lead to the alarm decision. These seven metrics will be used as the evidence or "cause" nodes of the BN. In this section, we will identify the relations between these nodes, which are represented in the form of the Bayesian edges. In order to do so, first, we will apply the understanding of the attack behaviours obtained from Chapter 3 to construct a general BN. We then use the acquired training data from Section 4.3.1 to study further relations between the nodes in this structure.

Given the evidence and the effect nodes in a Bayesian model, the easiest and most common constructed model is the naive Bayes. In this form, every evidence has an edge connected directly to the "effect" node, which indicates that all the evidences play same level influence to answer the query, and there are no direct relations between the evidences. This naive model for 6LoWPAN IDS is illustrated in Figure 4.4 below.

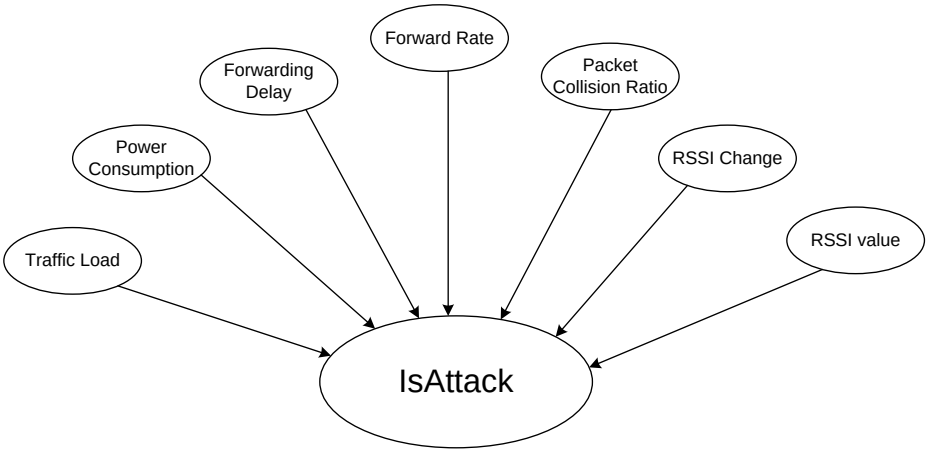


Figure 4.4. The naive Bayesian model for 6LoWPAN RPL

We do not use this naive Bayes for our model, because this model cannot reflect the insightful understandings of the threats, which were studied in Chapter 3. These understandings need to be integrated into the Bayes model to increase the accurateness and effectiveness. Moreover, the IsAttack node has 7 inputs, which will make its CPT to be too large. For example, assume that each input can have either 1 of the 4 states (i.e. High, Medium, Low, Abnormal), the total entries of the CPT that need to be implemented in the query node will be $4^7 = 16384$, which is too large and not effective.

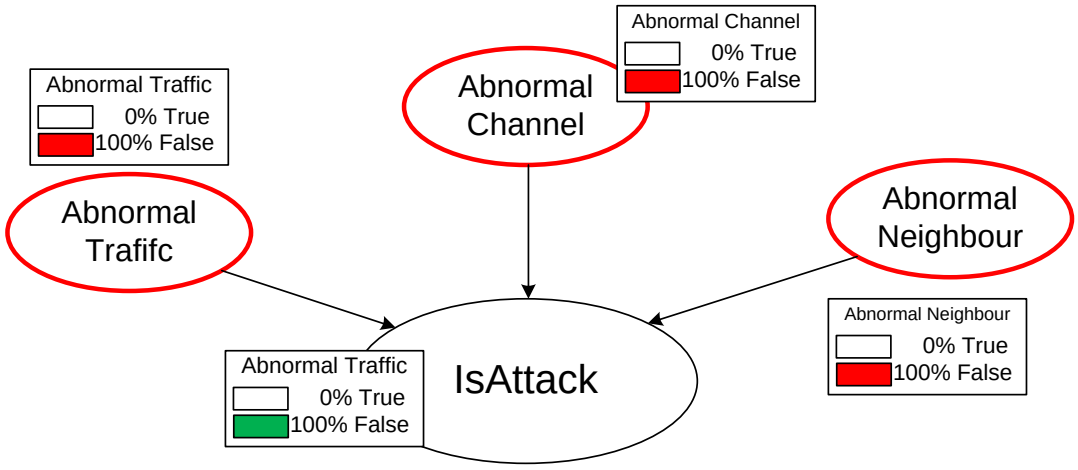
In Chapter 3, we have categorised the performance-type internal attack behaviours through three main types, including abnormal traffic, abnormal channel, and abnormal neighbour. Our Bayesian IDS assume that if a node is an internal attacker of the performance-type, their behaviours will be abnormal in at least one of these three categorisations, namely the traffic, channel or neighbour. On the other hand, if we can observe an abnormal behaviour in any of these categories, we can suspect the occurrence of an internal attacker following the performance-type.

The *IsAttack* query can be divided into the three smaller queries as follows:

- *Abnormal Traffic*: checking whether there is any abnormal behaviour in Traffic of the monitoring node
- *Abnormal Channel*: checking whether there is any abnormal behaviour in Channel around the monitoring node
- *Abnormal Neighbour*: checking whether the monitoring node has any abnormal behaviour in making other nodes to take it as a neighbour.

Before answering whether a behaviour is abnormal of the performance-type, the Bayesian model will first answer whether that behaviour falls in any of these three abnormal categories.

We construct three BN nodes for the three aforementioned abnormal categories as shown in Figure 4.5. Each node can have either True or False state. A "True" state indicates that the behaviour is abnormal, while a "False" state indicates a normal behaviour. If the input state of any of the Abnormal Channel, Abnormal Traffic, and Abnormal Neighbour is True, the state of Is Attack need to be True. On the other hand, the state of Is Attack node is only "False" when the input states of all of its parents are "False". All of these conditions are reflected in the CPT of *Is Attack* node, as presented in Figure 4.5 below.



Abnormal Traffic	True				False			
Abnormal Channel	True		False		True		False	
Abnormal Neighbour	True	False	True	False	True	False	True	False
True	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
False	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Figure 4.5. States and CPT of the query nodes

In Chapter 3, we showed the metrics that are sensitive to each of the sub-categories in the performance-type internal threats. Hence, for each of the sub-query, we will take only the metrics that are sensitive to the related category as its input. By reducing the number of inputs, we will reduce significantly the size of the CPT tables at the querying node.

According to Chapter 3, attacks that aim at node traffic will create abnormal change in the three metrics {*Forwarding Delay*, *Forwarding Rate*, and *Power Consumption*}. Therefore, whenever there are abnormal changes regarding these metrics, we can suspect the occurrence of traffic attacks. However, we also need to identify other events, which

can also create the changes at this traffic to eliminate any false decision. These events are analysed as follows:

- **Traffic load:** We see that the traffic load towards the node can be also another reason for such changes. For example, if a node has more traffic load, it will be more likely to have a higher forwarding delay, forwarding rate, and power consumption due to more work to be processed. Therefore, the Bayesian regarding the traffic module also needs to consider the Traffic Load metric to eliminate any false alarm of detecting an abnormal traffic behaviour caused by the high traffic.
- **Environment condition:** this factor can affect the network performance, for example, the weather condition can affect the signal of the communication, hence decrease the forwarding rate. The simulation tool available to us (i.e. Contiki and Cooja), however, cannot reflect properly this factor. The best representation they can provide is to set the transmission rate to be a fixed level during the whole simulation, which makes the variable constant all the time, so it does not reflect the reality as well as not helpful for the training. As a result, we did not include this factor in the study. However, this feature can be easily aware by the operators (e.g. they will know about weather condition or any environment incidents happens in the implementation area of the network), so we suggest that apart from the results given by the IDS, they can take this factor into account. For example, if the system raises the alarm while the current time has bad weather, or some incident is happening, the operator can assume this is created by environmental factors rather than by the internal attackers.
- **Other internal attack behaviours:** Chapter 3 showed that the channel attacks can create abnormal changes in forwarding rate as well as forwarding delay, neighbour attack can affect the forwarding delay, and topology attacks may create impacts to all of these inputs. However, these attacks will be reflected in different IDS modules that are directly related to them rather than including in this module. In case of the performance-type internal threats, the Bayesian IDS will give probability outputs for the state of each sub-query (for example: Abnormal Traffic 20% False, 80% True; Abnormal Channel 70% False, 30% True; Abnormal Neighbour: 100% False, 0% True) and the final query. In case the final query returns a positive output, we can look further at the sub-query outputs to see what types of internal attacks are more likely to happen in the network. On the other

hand, in case of the topology attacks, we will develop a specification-based IDS module in Chapter 5 to deal with, so we can skip them in this module.

Whenever the IDS observes evidences of abnormal performance regarding these metrics, it can give high probability output in concluding of a traffic attack. As a result, we can form this sub-query as shown in Figure 4.6.

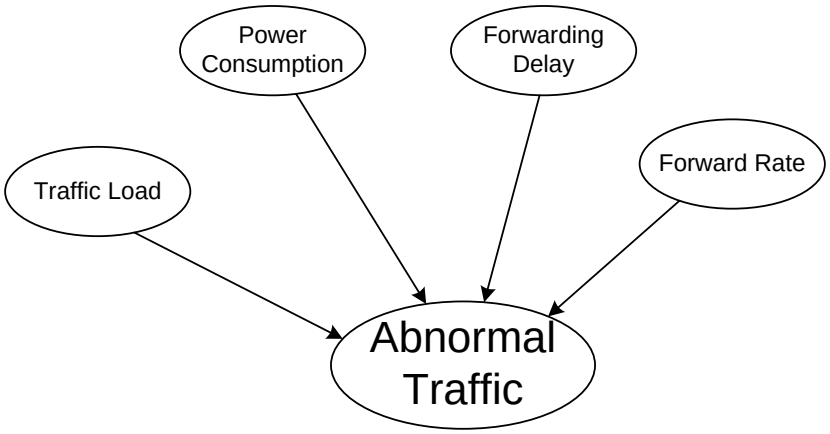


Figure 4.6. Bayesian model for the query of Abnormal Traffic node

Similarly, from the knowledge obtained in Chapter 3, we use the four metrics {*Power Consumption, Forwarding Delay, and Forwarding Rate, Packet Collision*} to predict the sub-query of *Abnormal Channel*; and we use the two metrics {*RSSI changes, RSSI value*} to predict the query of the *Abnormal Neighbour*. The Bayesian model for these two queries are given in Figure 4.7.

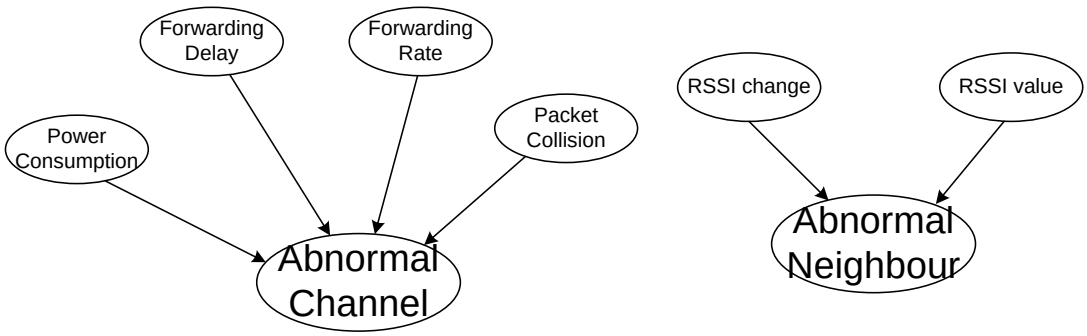


Figure 4.7. Bayesian model for the query of Abnormal Channel and Abnormal Neighbour node

The relations between the metrics can be derived from the data acquired from Section 4.4.1. In detail, we calculate the Pearson correlation between every pair of the input metrics based on the acquired time series data of the normal scenario to see the relations between them. The relations between the nodes need to be studied because such relations will help to identify better BN input. For example, if the monitoring node cannot properly collect the information regarding an input, it can predict this input based on information

coming from other input that has direct relation to that input. Note that we only consider data in the normal scenario because these relations were created by the normal network operating mechanisms. In case of attack scenarios, the attackers can manipulate the input data up to their choices; hence, any relations if detected cannot be explained by the real relations and therefore are not reliable.

The results of the Pearson correlations based on the acquired data in Section 4.4.1 are represented in Table 4.3. From the table, we only see three pairs with strong statistical correlation, which are the {*Traffic Load, Forwarding Delay*} (5% statistically significance), {*Traffic Load, Power Consumption*} (10% statistically significance), and {*Power Consumption, Forwarding Delay*} (10% statistically significance). The negative sign of the coefficient in the table indicates that the two metrics have inversely proportional relation, which mean the value of one metric increase will lead to the decrease of the other metric's value. Similarly, the positive sign of the coefficient shows the direct proportional relation of the two metrics.

Table 4.3. Statistical correlation between the variables in the normal performance scenarios (only show the coefficient of the relations with significant statistic (<10%) – the blue cells)

Variables	**Power Consmp.**	**Traffic Load**	**Forward Delay**	**Forward Rate**	**Packet Collision**	**RSSI Change**	**RSSI value**
Power Consmp.							
Traffic Load	0.314						
Forward Delay	-0.142	0.223					
Forward Rate	-0.149	-0.100	-0.447				
Packet Collision	0.378	0.252	0.508	-0.256			
RSSI Change	-[1]	-	-	-	-		
RSSI value	-	-	-	-	-	-	

We will assess further the cause-effect direction among these correlated metrics. In normal performance, the nodes need to receive a packet first before it can process and forward. The more packets that are sent to the node, the more power it needs to consume to process and forward them. Besides, when there are more packets sent to a node, the

[1] RSSI values recorded from a normal node are mostly constant through time; hence, there are no correlations with other metrics. Similarly, the RSSI change is almost 0 through simulation time, therefore, no correlations to other metrics can be calculated.

processing queue is longer so the time to forward the packets will be more delay. Hence, it can be considered that the traffic is the cause that creates the effect of the forwarding and power consumption behaviours. As a result, the Traffic Load metric will be the parent of the Forwarding Rate and the Power Consumption metric. On the other hand, the Traffic Load is not correlated to the Forwarding Rate because in the normal condition, nodes have the retransmit mechanism to make the dropping rate as low as possible, which tend to assure the delivery of the packets. Likewise, the Forwarding Rate and the Power Consumption are correlated because they both have the same cause, which is the Traffic Load. For this pair, we cannot identify which metric is the cause of the other. However, the correlation implies that if we know the state of a node, we can predict the state of the other node, so we will choose the metric, which is easier to acquire data as the cause. In this case, we choose the Power Consumption as the cause and the Forwarding Delay as the effect.

Combining the models in Figure 4.5, 4.6, 4.7, and the statistical correlations in Table 4.3, and the justification of the cause-effect relations, we construct the Bayesian-based model for 6LoWPAN RPL as shown in Figure 4.8.

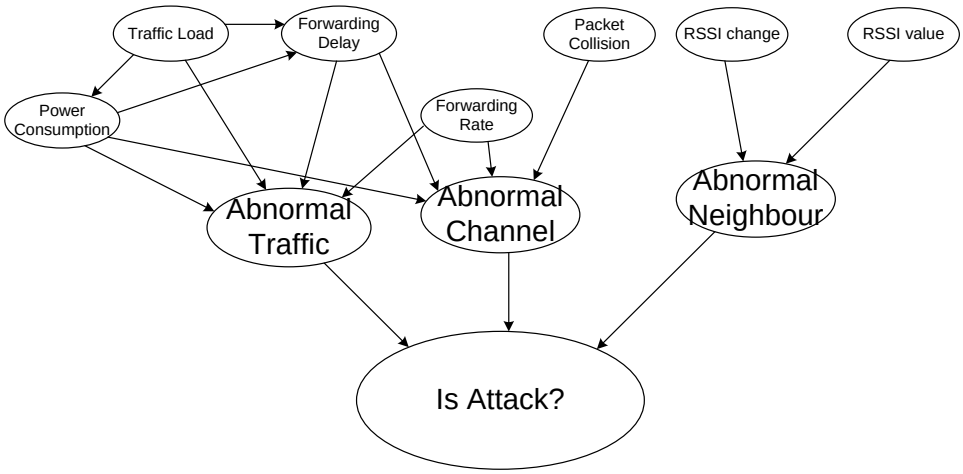


Figure 4.8. The structure of the constructed BN model

4.4.3. Parametric Learning

In this section, we present the method to set up the value for identifying the node states and the CPT using the acquired data in Section 4.4.1. Before using the acquired data, we need to eliminate all the low quality part that may falsify the model. In detail, the data regarding all the nodes' operation in the first minute does not reflect the real performance, because the nodes are in its setting up stage. Hence, we will eliminate this data. The filtered dataset will be used to set up the parametric for the Bayesian model.

4.4.3.1. Setting up the Node States

Setting up thresholds to convert the numerical data to the state-form is not an easy task. The reason is that when a node sitting in different network locations, it will have different performance pattern at each location. For example, the "High" traffic that observed in a border node may be considered "Low" for a node sitting near the sink. Defining specific threshold for each network node does not help, because during the operation, the changes of routing will lead to significantly changes in the performance pattern of the node (legitimately), while such changes cannot be obtained from the training phase. Besides, even when the threshold for differentiating Normal and Abnormal performance is set, there is no mechanism to consider a node of which behaviour are always near the "Abnormal" range. As a result, the attackers can adjust the behaviours so that they create impacts only at a certain level, which significantly affect the performance but not exceed the threshold to prevent the chance of being detected by the monitoring node. Although the impact in short-term is not a lot, attackers can create massive downgrade of network performance in the long-term while the IDS cannot detect them.

In this section, we first establish the states for each of the metrics. We then propose a method to set up the state thresholds, which create the range for each state and allow transforming any numerical data into state form. We also include a mechanism to punish the node if they have poor performance in long-term to detect the attacks, which lower the effect just to pass the IDS threshold.

For the five variables: *Power Consumption, Traffic Load, Forwarding Rate, Forwarding Delay, and Packet Collision,* we propose to have four states in each node, including *Low, Average, High,* and *Abnormal*. In a normal condition, a legitimate node only has its behaviours fall into either *Low*, *Average,* or *High* state. The *Abnormal* range is recorded in the attacked scenarios. Our method to define the node state is based on the distribution of the acquired data rather than based on the value.

Take the *Power Consumption* as an example, after obtaining these metrics from the training data set, we will divide them into two set $P_{normal} = \{P_{1n}, P_{2n}, \ldots, P_{kn}\}$ is the collection of power consumption value in the normal scenario simulation, while $P_{abnormal} = \{P_{1a}, P_{2a}, \ldots, P_{la}\}$ is the collection of that value in the attacked scenario. For the normal set, we will define the 2 states threshold $\{\alpha, \beta\}$ according to the distribution of the metric data. In detail, we set:

1. P_{low} is a value that less than α% of the acquired data on the P_{normal} set have smaller value than. As a result, any data of P falls into the [0, P_{low}] range will later be transformed to the "Low" state.
2. P_{high} is a value that less than β% of the P members has larger value than. As a result, any data of P that fall into the [P_{high}, P_{max}] will later be transformed to the "High" state.
3. Any data that fall into the [P_{low}, P_{high}] will be transformed to the "Medium" state.

In this particular IDS setting, we choose $\alpha = \beta = 15\%$, but these parameters can be adjusted to adapt when applying in other scenarios. The illustration of such division is given in Figure 4.9.

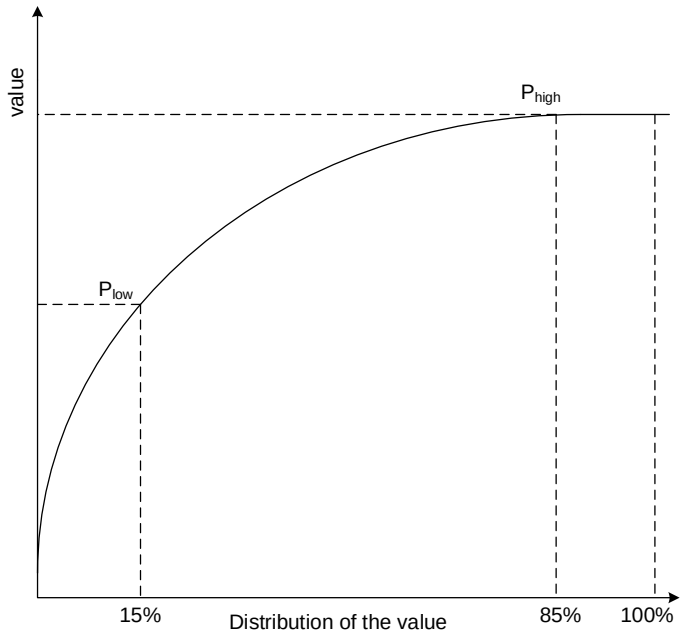


Figure 4.9. Division of Power Consumption variable into low, average, high states

We then use the abnormal data set to set up the Abnormal state as follows.

Let's call:

$P_{n\_max} = \max(P_{normal})$

$P_{n\_min} = \min(P_{normal})$

$P_{a\_max} = \max(P_{abnormal})$

$P_{a\_min} = \min(P_{abnormal})$

In case $P_{n\_max} < P_{a\_min}$ then there is no intersection between the P_{normal} and $P_{abnormal}$ set, and $P_{abnormal}$ is set to P_{a_min} can help to easily identify the normal and abnormal power consumption. This case is illustrated in Figure 4.10(a).

On the other hand, if $P_{n\_max} > P_{a\_min}$, there will be an intersection between P_{normal} and $P_{abnormal}$. This case is illustrated in Figure 4.10(b). In such case, if an observed value is in this intersection, we cannot decide whether it is a high normal or a low abnormal value. If we detect it as an abnormal value, then the chance for false detection is high. However, if we detect it as a normal value, attackers may take advantage of this by lower the short-term impact to create long-term damage. In order to overcome this issue, we propose to add an abnormal counter to each metric, which will increase by 1 every time the node's performance value fall into the intersection area. If this counter reaches a counter threshold, then the value will be set to the abnormal state. If a value is observed at the abnormal state, there will be a higher chance that the monitored node is under attack.

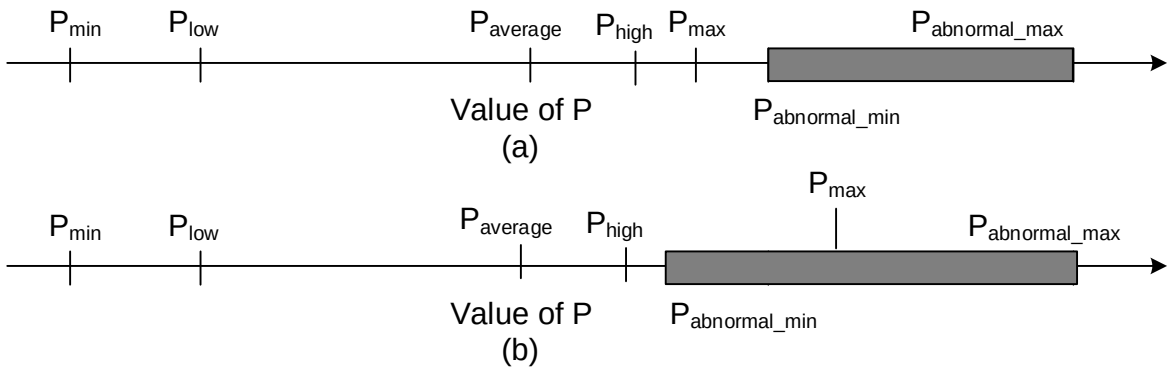


Figure 4.10. Example of setting state for a Bayesian variable given the normal and abnormal data set

We do similar state setting for other variables. For the Forwarding Rate, the setting up threshold is opposite to other node, because the lower the value, the more chance will display anomalous behaviour. On the other hand, the RSSI Value will have two states; the Normal state indicates that this value is in the normal range according to the device specification, while the Abnormal state indicate that this value is out of the normal range. Similarly, the RSSI Change also has two states including Normal and Abnormal. The Abnormal state indicates that there is an abnormal change of the RSSI Value of a node, in which the deviation of the RSSI Value between two times are larger than a certain threshold.

After setting up the thresholds, we can translate the data into the state data. For example, the data in Table 4.2 can be transformed into state data as shown in Table 4.4 below.

Table 4.4. Example of state transformed data acquired from a node during time

Metrics/time	**Period 1**	**Period 2**	**Period 3**	**Period 4**	….
Forwarding rate (%)	High	High	High	Abnormal	
Forwarding delay (ms)	Low	Low	Medium	High	

Power consumption (%)	Low	Low	Medium	High	
Traffic load (no of packets)	Low	Low	Low	Low	
Packet collision ratio (%)	Low	Low	Low	Low	
RSSI change (%)	Normal	Normal	Normal	Normal	
RSSI value (dBm)	Normal	Normal	Normal	Normal	
ATTACK	**FALSE**	**FALSE**	**FALSE**	**TRUE**	

4.4.3.2. Setting up the Conditional Probability Table

The last sections of this chapter provide a method to transfer the data set in the numerical-form into state-form and the BN structure. We use the SamIam, a common applied BN tool [95] to construct the BN with the formed structure, and train the parametric through the SamIam's EM learning function with the acquired state-form data. The EM learning algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. This algorithm is used widely in setting up the BN with training data. After the training, we will have a CPT set up for every query and sub-query node in the BN. We apply the BN model with the acquired parametric to the whole state-form data set to filter the inaccuracy answer, i. e., the False Positive (FP) and the False Negative (FN) cases. High frequency inaccuracy cases are adjusted by the Sensitivity Analysis function providing by SamIam, for instance, putting the constraints to increase the probability of Positive in the answer of the query in FP cases with malicious behaviours detected as benign or to increase the probability of Negative in FN cases with benign behaviours detected as malicious. After adjusting the parametric, we have a BN that is ready to answer the query with the corresponding input data.

4.4.4. Implementing the Constructed Bayesian Model

After the steps from Section 4.4.1 to Section 4.4.3, we have constructed a theoretical Bayesian-based IDS model that can take the statistics of the performance data, transfer them into Bayes nodes' state, and give the probable answer to the IDS query. In this section, we will present the detailed architecture of the monitoring IDS for the real network operation based on the design in Section 4.2, where we suggested the use of additional interface to transfer the IDS data between the monitoring nodes and the IDS server through a base station. Examples of the additional interfaces to be used are Wi-Fi, 3G, or GSM, which allows high speed communication, large coverage, with low cost of implementing and operating.

We will consider the distributing options as reviewed in Section 2.6.4. For the network-based approach, placing the monitoring module at the sink is not suitable because a huge amount of IDS data will need to be transferred to the sink, which will create both the overhead and storage issues. On the other hand, in host-based approach, asking every node to send the IDS data to the server means that every 6LoWPAN node needs to have an additional interface, which is not necessary. The hybrid approach based on cluster monitoring seems to be the most suitable for several reasons as follows. First, only the cluster head nodes need to have the additional interface to communicate with IDS server so it will save the resources. We can plan the situations of such monitoring nodes in the network so that they can cover the whole network easily. Second, local work in each cluster is not much and manageable for the cluster head. As a result, there will be only a small amount of overhead generated.

The detailed designs are as follows. Besides the sink, the network will consist of two types of nodes, the monitored nodes are with normal Contiki-firmware and the monitoring nodes are with modified Contiki-firmware to record the relevant messages from the monitored nodes and extract the IDS metrics. The monitoring nodes are provided an additional interface so that it can send the extracted data to an IDS server through a base station. The Bayesian learning and monitoring process is done on the server side. For learning, the server will collect all the data from its monitoring nodes and do the parametric setup as presented in the previous sections. For monitoring, it will feed the data from a particular node in a specific period into the constructed BN and give the output through the three modules as shown in Section 4.3.4. Depend on the topology design, the monitoring nodes are distributed in the network so as they can cover the whole area, while each monitoring node can form a cluster of around 12 nodes for managing the workload. The monitoring node is set as the cluster head and nodes inside its coverage are the cluster members. Each node can belong to more than one cluster. When such case happens, there will be more information from the monitored node sent to the server, so it can cross check to verify further.

Because all the cluster members are the neighbour of the cluster head, the head can record all the broadcasting control messages and the unicast message toward itself. We set the cluster head to listen to the data packets that go in and out of each of its cluster members. By analysing the source and the destination in these packets, we can identify the *Forwarding Load* (number of packets that need to be forwarded from the monitored nodes), *Forwarding Delay* (the average time since a packet was sent to a monitored node

to the time it was forwarded), and *Delivery Rate* (percentage between the number of packets which were sent to a monitored node and the number of packets which were forwarded out of this node). The RSSI value is recorded through extracting the packet property every time the cluster head receives packets from its members. Through the recorded RSSI of the cluster members, the cluster head can derive the *RSSI Value* and *RSSI Change* data. The only missing metric is the *Power Consumption*; in which we will modify the cluster member code further to make them report this metric to the cluster head periodically. The value of the period time is set up depends on the particular scenario.

Our design can provide several advantages as follows. First, the cluster head does not have to run heavy computation from IDS algorithms. As a result, it can save resources and have longer life. Second, there will be much less overhead from the IDS because the IDS data is transferred by different communication channel. Moreover, the speed of IDS data transfer will be much faster, which allow to make the decision quickly when detecting the malicious source. Third, the complexity of the IDS algorithms can be extended to further improving the accuracy because the computing capability of the server will be much stronger than that of a sensor node. Fourth, it can expand the scale of the system due to generating only little overhead.

In the next section, we will present a case study of applying the whole process as presented in this section in a 6LoWPAN scenario to evaluate the effectiveness of the solution.

4.5. A Case Study of Building and Testing the Bayesian-based Module

Our constructed Bayesian structure from Section 4.4 is obtained from the understanding of the performance-type attacks and network behaviours, so it can be remained for every scenario. However, its parametric values (i. e. node probability and CPT) need to be trained every time before applying to a specific situation.

In this section, we present a case study of constructing the BN from start until the testing phase of the detection effectiveness of the system.

4.5.1. Simulation Set up

We do not reuse the scenario in Chapter 3 for evaluation in this chapter because we create that scenario only for easily detecting the sensitive metrics to the performance-type attack and their trend. It also does not take into account the form of clusters with the cluster heads to monitor the cluster members. In this chapter, we set up a larger and more complex network with the appearance of cluster heads. We assume that the cluster heads will send the IDS traffic that they record from their cluster members to a server through an additional channel (i. e. Wi-Fi, 3G, 4G, GSM). The server will have the Bayesian code implemented with three modules as presented in Section 4.3.4. It will judge the IDS data for each of the cluster member nodes in the network. The cluster heads are assumed to have higher security protection so as their monitoring data of other cluster member are reliable. On monitoring, the cluster heads are set to listen to the data packets that go in and out of each of its cluster members, while other members will report its power consumption to the cluster head every one minute. Cluster heads will process the data that they observe from the sensor nodes to extract the seven metrics as justified in Section 4.4.1.1. As the implementations of multiple interface sensors are not yet available in Contiki, we run the detection process through analysing the simulation trace files as the proof-of-concept.

Our application scenario consists of 100 nodes placed in a 600x600m area; each node has a transmission range of 50m. The time to run a simulation with Cooja Contiki is long (a simulation with the setup describe in this chapter in our computer (Ubuntu 13, i7 3720QM processor, 24GB RAM) took about 8 hours to complete). Due to a large number of simulations that need to be done and limited time and resource, we could not simulate with larger number of nodes. We placed the IDS cluster heads manually so that each head can cover from 8-10 cluster members. We did not spread the nodes all over the network, because that will shorten the largest number of hops towards the sink. Instead, we set up the cluster heads to form the two backbones 29-13-32-42-60-46-84, and 29-13-32-17-44-72-98. By doing so, we can measure the effect of the attacks with more number of hops, more traffic with the same number of nodes. The topology set up is shown in Figure 4.12 below. There is one sink placed in the top left (the green node in Figure 4.11) and 11 IDS cluster heads (the yellow nodes) to cover the operation of the remaining 88 nodes in the network (the sky blue nodes). The connectivity map of the network set up is expressed in Figure 4.12. Every node sends packet to the sink at the rate of one packet every 60 seconds for 31.5 minutes.

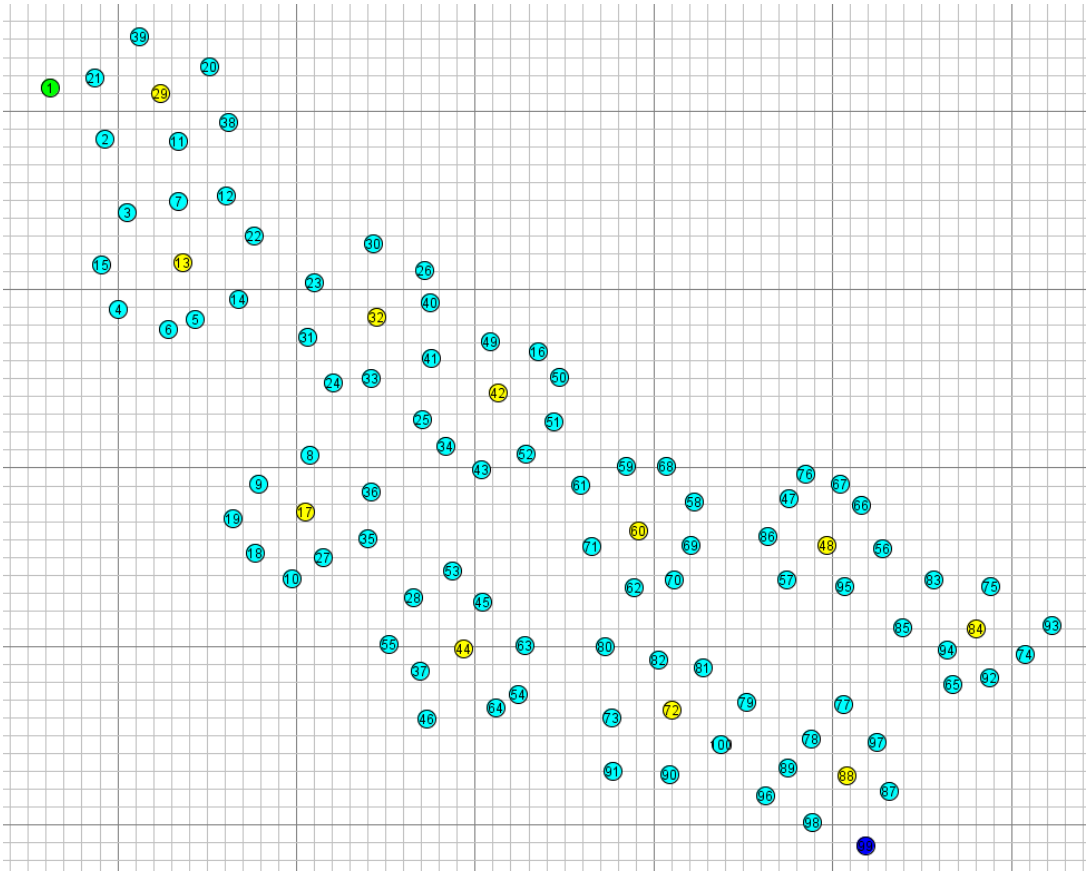


Figure 4.11. Topology set up for testing BN module

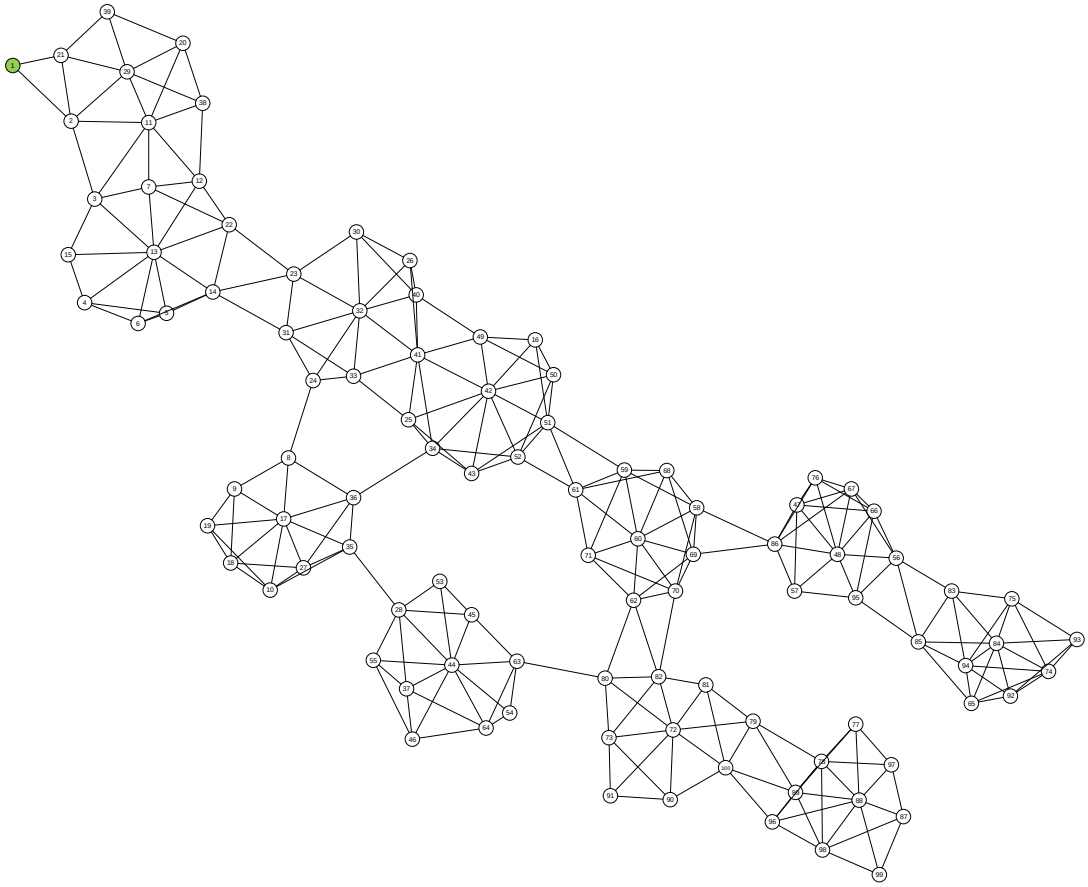


Figure 4.12. Connectivity map of the set up topology

We run 10 different simulation scenarios for the normal network condition with different simulation seeds to generate normal performance data set. For each of the attacks, including Black Hole, Grey Hole, Delaying, Jamming type 2, and Hello Flood attack, we implement the attacks in two random positions to simulate and collect the anomalous data set. Each of the attacking node will initiate the attack after the network starts for 1 minute. The detail parameters set up for those scenarios are summarised in Table 4.5 below.

Table 4.5. General set up for collecting training data set for Bayesian-based module

Attack type	Parameter set up
Black Hole	The dropping rate is set to 100%
Grey Hole	The dropping rate is set to 20%
Delaying	The adding delay is set to 2 seconds
Jamming type 2	Period of broadcast dump packet = 15 seconds
Hello Flood	RSSI of the attackers is set to -10 dBm

4.5.2. Numerical-state Reference Set up

Following the state discretions method presented in Section 4.4.3.2, we achieved the following set up for each parameter in Table 4.6.

Table 4.6. Threshold set up for each parameters in the Bayesian module

Variables	Low	Medium	High	Abnormal
Forwarding rate (%)	80-90%	90-95%	>95%	<80%
Forwarding delay (ms)	<200	200-1000	1000-3500	>2000
Power consumption (CPU + transmit)	<181818	181818-236415	236415-384019	>352522
Traffic load (number of packets)	<3	3-44	44-93	>93
Packet collision rate	<5%	5-15%	15-30%	>30%
RSSI value	Normal value: -70 to -30 dBm		> -30 dBm	
RSSI change	Normal: not different more than 20% through time		Change more than 20% Two nodes have the same RSSI value	

For RSSI changes, each cluster head node will record the corresponding RSSI value for each of its neighbours and check if the RSSI value is changed significantly through time (20% difference), or check whether the RSSI value is the same for the two neighbours.

We then transform the data set collected from Section 4.5.1 into state-based data. The data will have the format like the examples given in Table 4.7 below.

Table 4.7. Examples of state recorded

Forwarding Rate	Forwarding Delay	Power Consumption	Traffic Load	Number of Collisions	RSSI changes	RSSI values
High	Low	Low	Abnormal High	Abnormal High	Normal	Normal
N/A	N/A	Low	N/A	Low	Normal	Normal
High	High	High	Medium	Low	Normal	Ab-normal

4.5.3. Evaluation Results and Discussion

4.5.3.1. Detection Efficiency

For each of the attacks, including Black Hole, Grey Hole, Delaying, Jamming type 2, and Hello Flood attack, we implement the relevant attack code in a random position of 88 normal senders. The attacks are initiated after 1 minute when the topology establishment phase is done.

We consider the set of states collected at the cluster head for each monitoring node every 1 minute is a test input. We know the result of this input because we already know the position of the attacker, so if this input belongs to the attacker, the result is TRUE and vice versa. The Bayesian-based IDS will judge this test input based on the training data as in Section 4.5.1. We set the threshold for alarming an attack is when Bayesian query return the TRUE state with a value, which is more than 80%. We will match the judgment of the Bayesian-based module and the result from the testing data to see the effectiveness of the system.

Table 4.8 below shows the TPR and FPR of the Black hole attack, Grey hole, Delaying, Jamming attack and Hello flood attack.

Table 4.8. TPR and FPR of Black Hole attack, Grey Hole attack, Delaying, Jamming attack and Hello flood attack

	TPR(%)	**FPR(%)**
Black Hole	100	0
Grey Hole	90	0
Delaying	100	4.2
Jamming	86.67	9.39
Hello flood	100	0

For simplicity, from now we call *Power Consumption, Traffic Load, Forwarding Delay, Forwarding Rate, Packet Collision, RSSI value,* and *RSSI change* as *PCon, TLoad, FDelay, FRate, PCol, RValue,* and *RChange* respectively. The state of the BN nodes including High, Low, Medium, Abnormal High, Abnormal Low, Normal, and No Information will be represented as H, L, M, AH, AL, N, and N/A respectively. When we mention a sample of the data set in the state form, states in such sample will belong to the BN nodes following this order [*Power Consumption, Traffic Load, Forwarding Delay, Forwarding Rate, Packet Collision, RSSI value, RSSI change*]. For example, a sample of

[H, L, H, L, A, N, N] means that the state of *Power Consumption* is High, *Traffic Load* is Low, and so on.

For the Black hole attack, our BN module not only shows a high TPR at 100%, but also does not raise false alarm with the FPR at 0%. The detection of the Black hole attack is straightforward and many other IDSs can reach the TPR of 100% just by checking the dropping rate of the monitored node. However, our Bayesian judges based on the view of multiple metrics, therefore, even in the case the state of *FRate* is AL, it may still reserve the alarm decision. For example, with the input [L, L, N/A, AL, L, N, N] the *IsAttack* answer is 75.59% True, which is under the 80% threshold, so there will be no alarm. This setting helps to reduce the FPR when compared to other IDSs by preventing the false alarm in case that the *FRate* is very low, but it is not the attack because the actual number of dropping packets is also very low due to a low *FLoad*. Figure 4.13 indicates the BN for a sample recorded at node 23 (the position of the Black hole attacker) at minute 3, which have input as [L, M, N/A, AL, L, N, N]. The *FDelay* cannot be measured because all the packets go through that node were dropped. However, with the Medium *FLoad* and Low PCon, the model can estimate it has 60% chance for the FDelay state to be Medium. The calculation of the Abnormal traffic node returns 88.13% that this record is an abnormal traffic behaviour, which encourages the system to raise the alarm (which is true) with 91.6% confidence. On the other hand, the *Abnormal Low* Forward Rate does not lead to the conclusion of Abnormal Channel behaviour (68.71% true) because the

Packet Collisions state is Low. Therefore, if we trace back to the sub-query prediction result, we will see the main cause and the type of the abnormal behaviour.

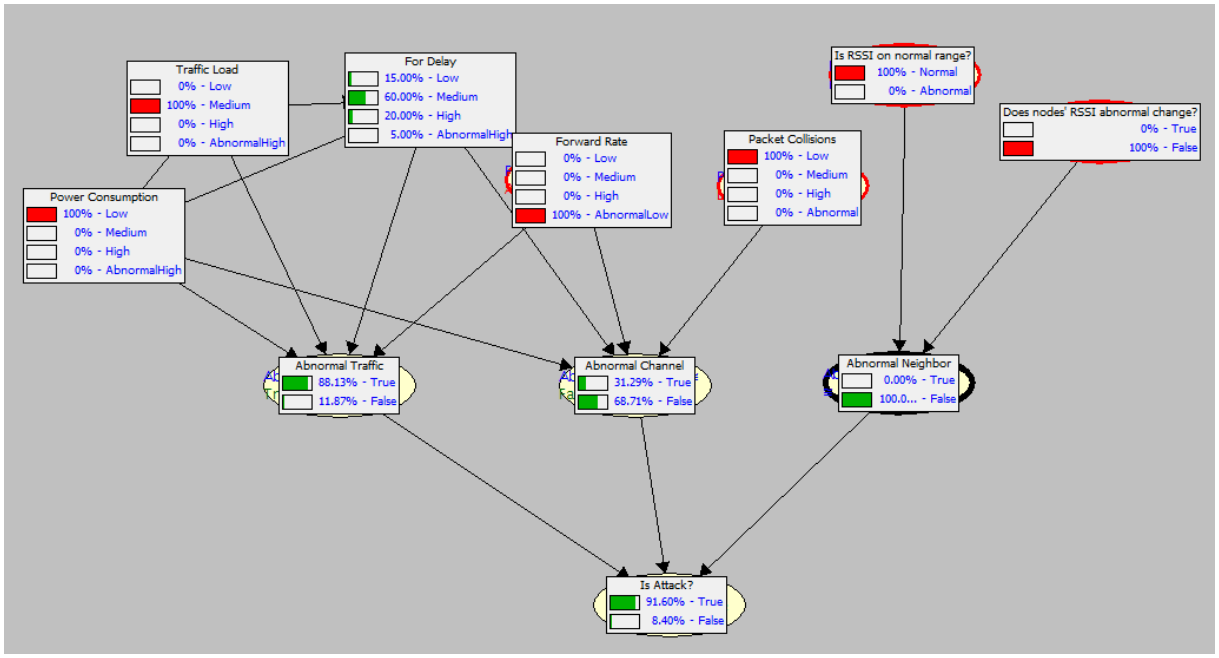


Figure 4.13. A sample of the Black Hole attack

Regarding the Grey Hole attack, the module achieved a 0% FPR, which can be explained similarly as explaining in the case of Black Hole attack. On the other hand, the TPR is at 90%. Checking the recorded states of the *FRate*, we observed that it flipped between L and N state at the first 3 periods, and since period 4th, it is stick with the AL state. This is because counting bit as presented in Section 4.4.3.2 was activated after 2 times the dropping rate falls in the intersection of the L and AL states. For the first 3 samples, the module cannot detect the attack, hence the judging results when the *FRate* node flipped between these two states are significantly different. Figure 4.14 and 4.15 show the records of the Bayesian node at minute 4 and 5 respectively. The only difference between these two records is the state of the *FRate* metric. If this state is L, the Abnormal Traffic node value is 71.38%, which make the query result to be 77.91%, not enough confidence to raise the alarm. On the other hand, if this state is AL, the Abnormal Traffic node value is 85.05%, which make the result to be 89.13% and raise a true positive alarm. This example shows the advantage of using the counting bit of the states, because if we do not use this bit, the Grey hole node's Forwarding Rate state will continue flipping between the L and

AL value, making the decision flipping from the Abnormal/Normal answer, and will lower the TPR.

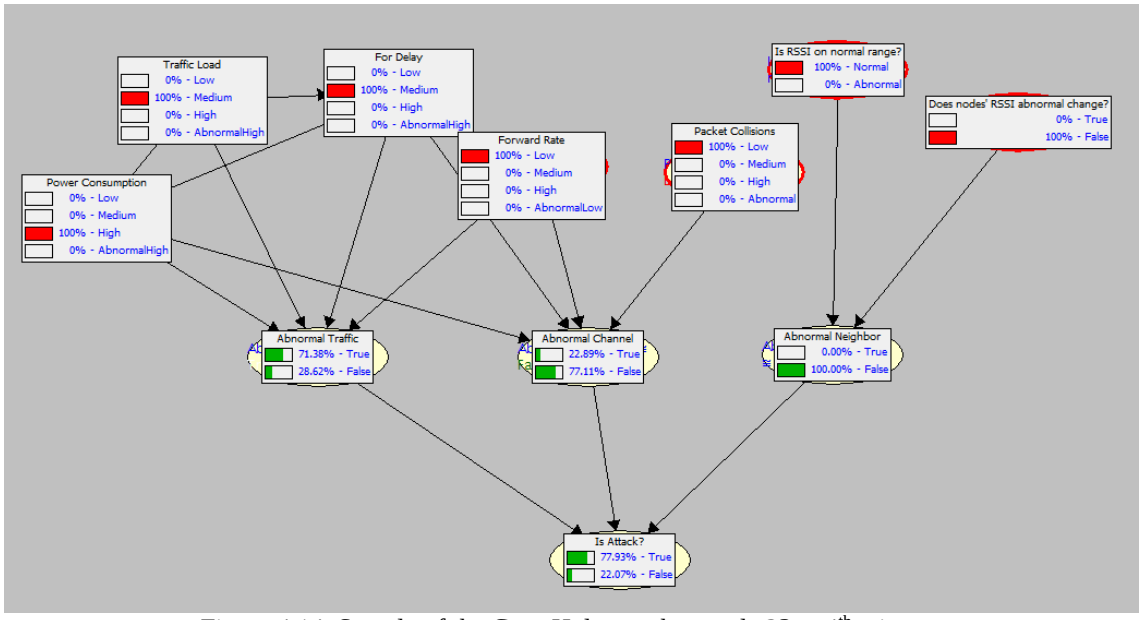


Figure 4.14. Sample of the Grey Hole attack at node 23 at 4th minutes

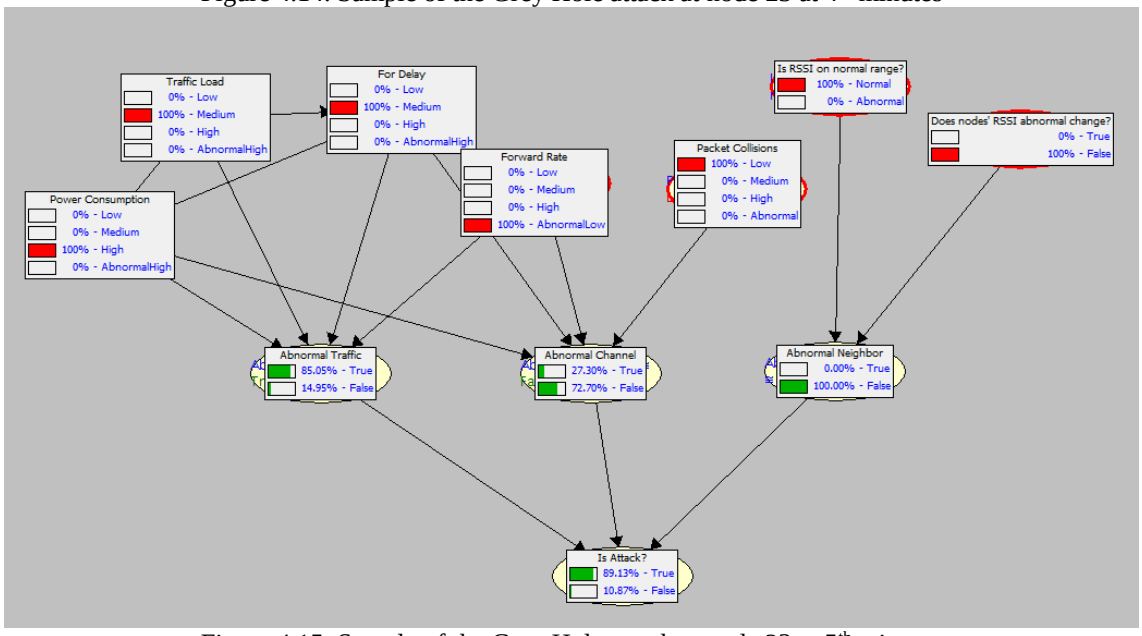


Figure 4.15. Sample of the Grey Hole attack at node 23 at 5th minutes

Regarding the Delaying attack, we achieved a TPR rate at 100% with 4.2% of FPR. For the Delaying attack, the observed *FDelay* state in the attacker is always AH, so the BN module can detect malicious behaviours accurately. For example, Figure 4.16 shows the record of a True Positive sample of node 23 (attacking is also at node 23). It can be seen that an AH state in *FDelay* has made the *AbnormalTraffic* to 81.88%, which makes the *IsAttack* query raise alarm with 92.45% confidence. Besides, it is also important to note that the *AbnormalChannel* node also rose to 58.32%, regardless of a *Low* state in *PCol*. This reflects the insightful understanding with which the system was implemented that *FDelay* and *FRate* also can be seen as the symptoms for Abnormal Channel attack.

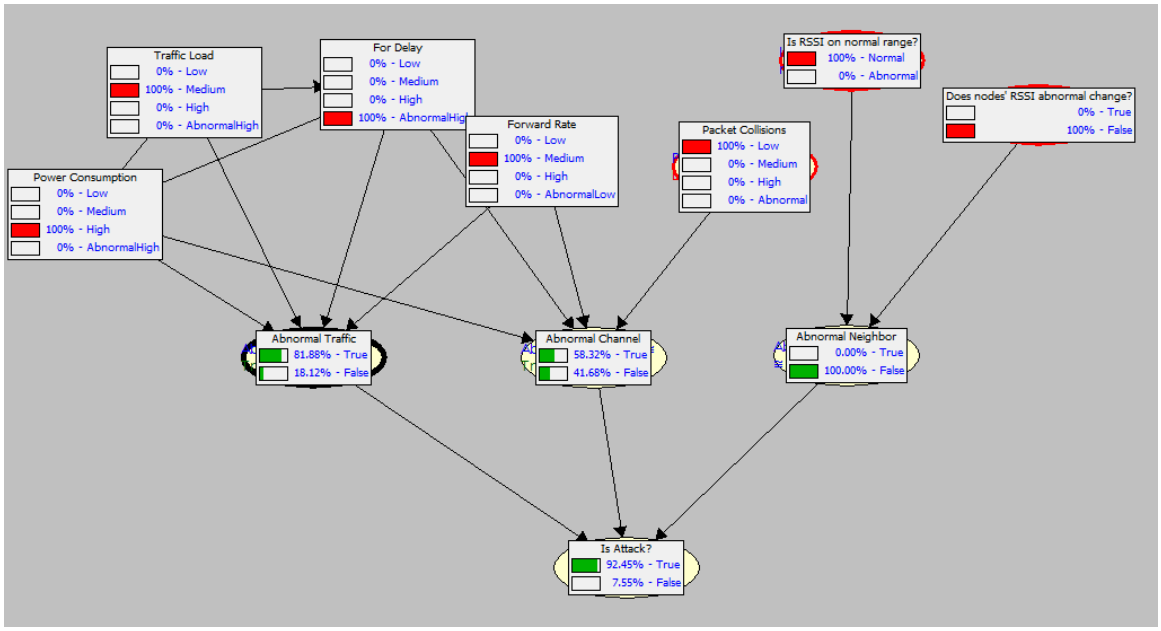


Figure 4.16. True positive Record of the Delaying attack at node 23

Figure 4.17 shows a False Positive case where a normal node behaviour is detected as abnormal. This is node 14, which has AL state in *FRate* metric because of dropping packets from node 23. When a node attacks the traffic of other nodes, the victims' behaviours will be affected significantly. Because the anomaly IDS system only judge the variables regarding the nodes' performance, such FP cases are difficult to be avoided. However, after detecting the malicious sources, if we can remove them to eliminate their impacts to the other nodes, the FPR will decrease significantly.

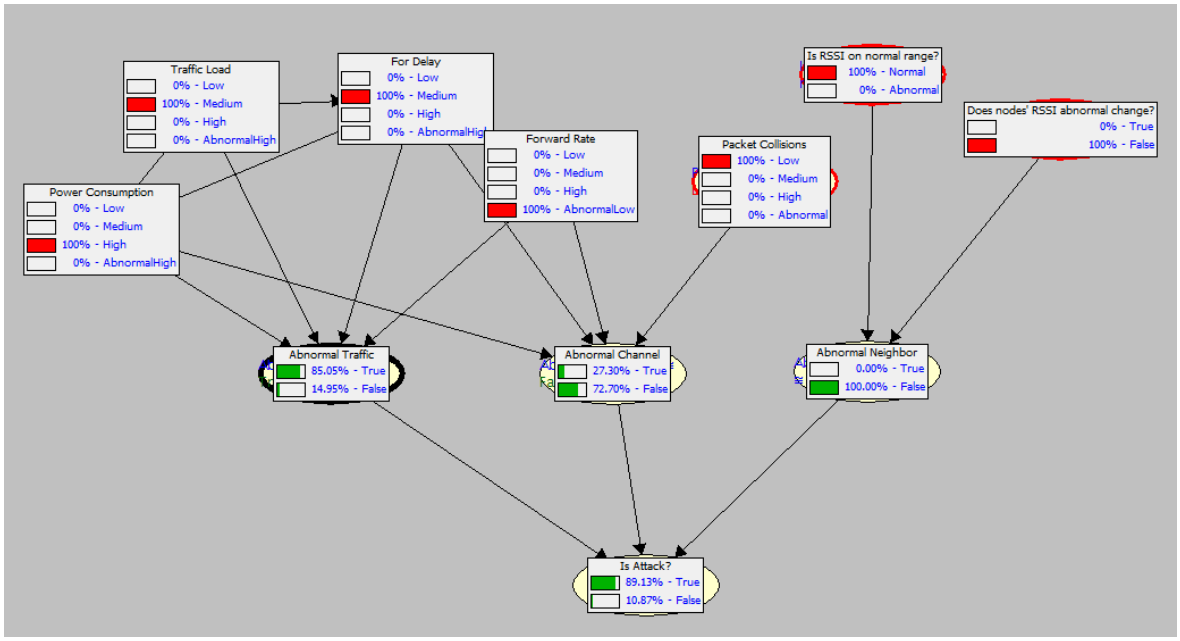


Figure 4.17. False positive record of the Delaying attack at node 23

Regarding the Jamming attack, we achieved a TPR of 86.67%, while the FPR is at 9.39%. We get a high FPR because of similar reasons as justified in the Delaying attack. In detail,

the behaviours of the Jamming attack node and the other nodes around it are quite similar, considering the metrics chosen in our Bayesian model. For example, the *PCol* of the area around the attacker is always high because the channel is manipulated during the attacking time. The *FDelay* and the *FRate* of nodes in this area are also affected by the Jamming behaviours, which increase the traffic in channel through sending dumb packets. The Bayesian model is not very effective in detecting the Jamming attacker, however, the trend of unstable performance of the area around help the system to spot the anomaly created by the Jamming attacker accurately. Once the affected area was pointed out, we can do further analysis to detect the attacker, for example, the attacker can be predicted as the node which is in the centre of the detected abnormal set.

Regarding the Hello Flood attack, we achieved an ideal TPR rate at 100% and 0% of FPR. This is because in this attack, the attacker is always detected because of having an out of normal range RSSI. In detail, the BN module always raises alarm with any input where *RValue* or *RChange* is with Abnormal or True state.

The use of RSSI in our simulation has a limitation, in which the Contiki-Cooja does not model the ways that RSSI of a node change through time. As such, node RSSI is a constant value, which make the IDS module detect the attack more easily. Applying a proper RSSI changing model will allow to see the how the IDS deal with the changes, however, we believe that the effectiveness of the IDS will not be decreased when such changes are applied. This is because in reality, if checking in a small sampling period of the IDS data, the RSSI changes are minor, which make the RSSI change and anomaly values will be significant and detected by the IDS easily.

Similarly, attacks towards neighbour ID like Sybil attack, Clone ID attack, and Neighbour attack will be detected through the same RSSI checking mechanisms. An illustration of the BN for the Hello flood attack is shown in Figure 4.18 below.

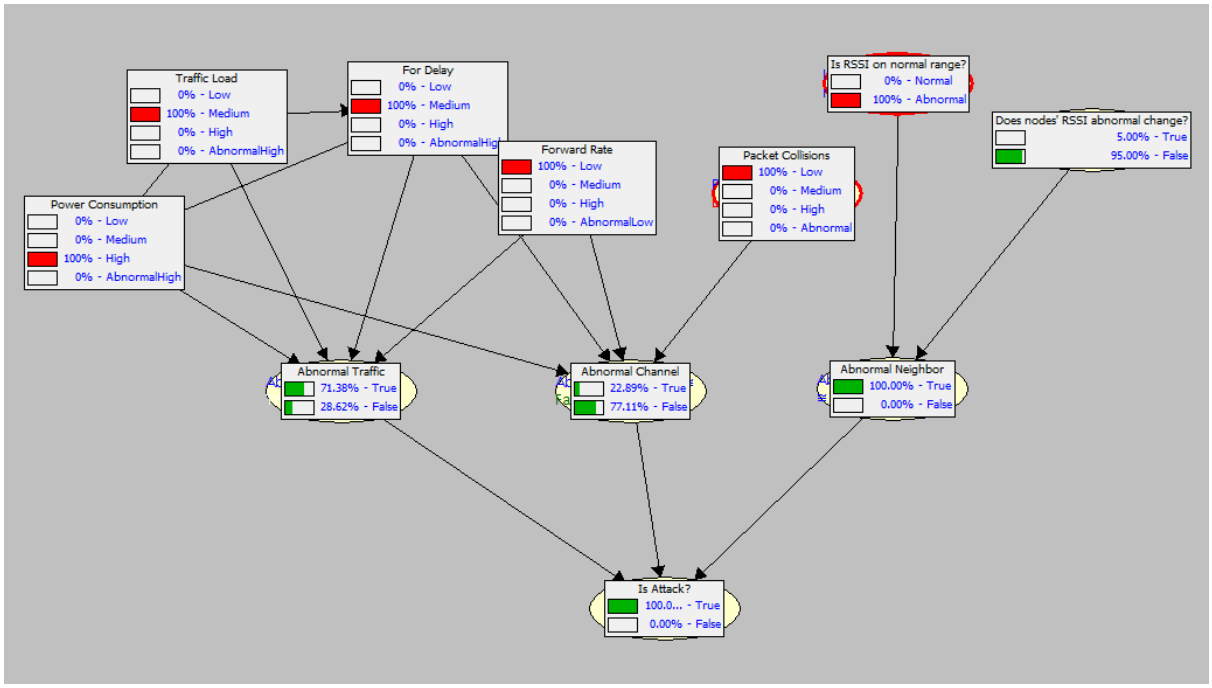


Figure 4.18. Record of the Hello Flood attack

4.5.3.2. Energy Efficiency

We measure the IDS overhead only in the Contiki side. Computing overhead created by putting the real IDS work on the IDS server through additional interface communication does not need to be evaluated because of the extensive capability in the server side given the cloud computing. There will be more power consumption at the cluster head created by using the additional interface to send the IDS data to the server, however, such power consumption is insignificant because the size of the data is small given the data are pre-processed to extract only the relevant metrics, while communication in additional channel does not take much power.

We ran the simulation with and without our IDS integration and obtained the energy and power consumption as calculated in formula (1) and (2) in Section 2.6.5. In normal RPL network, the energy is around 192J, while in the RPL with IDS integration, the average energy consumption is about 197J, which represents an increase of 2.6%, which is at acceptable and insignificant level. As a result, we can conclude that the module is energy efficient and can be scalable.

4.6. Chapter Summary

This chapter looked at detecting the performance-type internal threats, which aiming particularly at the traffic, channel, or neighbour. The considered attacks include the Black Hole, Grey Hole, Delaying, Jamming, and Hello Flood attack. We have presented a four-

step framework to construct a BN to apply in 6LoWPAN. In detail, we identified the seven metrics, which are used as evidences to judge a node's behaviour. The metrics include the *Power Consumption, Traffic Load, Forwarding Delay, Forwarding Rate, Packet Collisions, RSSI Change* and *RSSI Value*. We proposed the counting bit for each metric in the BN as a way to judge the node behaviour in long-term. For every scenario, our IDS requires a training phase that involves the collecting of simulating trace data from both the normal and the attacked scenarios. We have implemented a statistical Bayesian module to calculate the probability of the attacks given the input states of the Bayes nodes. The testing results show that this system can effectively detect the performance-type attacks with high TPR and low FPR in most of the attacks. The accurateness of the IDS can be improved further if additional mechanism is developed to eliminate the impacts of the malicious source after being detected. We also showed that our IDS architecture is energy efficient and well scalable.

In Chapter 3, we have pointed out that the anomaly-based IDS is not very effective in detecting the topology attacks. The Bayesian-based IDS in this chapter is not an exception because the statistical monitoring does not reflect well the detail of protocol operation. We believe that a proper profiling of the routing protocol operation will be needed for constructing a specification-based IDS for detecting this type of attacks. Without protocol specification, both the TPR and FPR of the IDS will be worse because the system cannot differentiate between the attackers and the victims of the attackers, of which behaviours may be at the same level of anomaly due to the attack impacts. Therefore, in the next chapter, we will specify the RPL operation and construct a specification-based IDS from this profile to deal with this topology attack. This specification-based module together with the Bayesian-based module in this chapter will help the system to detect most of the typical internal attacks presented in this thesis.

CHAPTER 5. A SPECIFICATION-BASED IDS FOR DETECTING 6LOWPAN TOPOLOGY THREATS

5.1. Introduction

In the last chapter, a Bayesian-based IDS was developed to detect the performance-type internal threats by judging node behaviours through evidences at traffic, channel, or neighbour using the statistical data from the seven metrics. Those evidences, however, are not very effective when judging the topology attackers. This is because the metrics mainly focus on the local performance of the nodes, while locally; the topology attackers do not perform any worse than they should. Moreover, the topology attackers break the optimal topology, so their neighbours are likely to have the same performance pattern as them. Therefore, these nodes and their neighbours are likely to receive the same judgements. This will make either the FPR of the Bayesian-IDS or any anomaly-based IDS to be high. Therefore, another separate IDS module needs to be developed to deal with the topology attacks. Because 6LoWPAN uses RPL as its underlying protocol, the specification-based IDS in this chapter indicates the specification of RPL operations.

The *specification-based IDS* detects the attackers' behaviours if they do not follow the expected behaviours as specified. This approach has the advantage of employing the knowledge of routing protocol to detect the illegitimate behaviours quickly. So far, specification-based detection has been applied to privileged programs, applications, and several sensor and ad hoc network protocols [68, 96]. There are also several RPL specifications [83, 97] but they are either in the conceptual form or are being developed for a different purpose than detecting the 6LoWPAN topology attacks.

In this chapter, we will develop, implement, and verify a specification-based IDS for RPL to detect the topology attacks in RPL-based network. Unlike a conceptual RPL specification built based on specialist knowledge in [83], this chapter develops a practical RPL operation model based on a semi-auto profiling technique, which then can be integrated to the monitoring agent to detect the topology attacks. Briefly, our approach involves the use of simulation trace files to generate an Extended Finite State Machines (EFSM - a Finite State Machine with statistical information about the transitions and states) for RPL. The technique of analysing the trace files to form the EFSM, which was inspired from [98], will help to quickly build a reliable profile compared to doing it

manually. We show how to implement this specification-based model to the network with a cluster architecture as applied in the previous chapter. We illustrate that our algorithm can effectively detect most of the typical topology attacks in real-time and with small amount of overhead.

The structure of this chapter is as follows. Section 5.2 discusses the related works that aim at securing the RPL, while Section 5.3 describes our specification-based approach to form the RPL profile. Section 5.4 presents the evaluation of the proposed solution and giving discussion.

5.2. Approaches to Secure the RPL

Initial work on securing RPL was done by Tsao [99], which focused on protecting RPL control messages (DIO, DIS, and DAO) as well as the routing information in IPv6 Hop-by-Hop Option Header and Routing Header. The authors suggest that RPL security objectives should be:

1. Participants of the DIO, DIS, and DAO message exchanges are authenticated.
2. The received DIO, DIS, and DAO messages are not modified during transportation.
3. The received DIO, DIS, and DAO messages are not retransmissions of previous messages.
4. The content of the DIO, DIS, and DAO messages may be made legible to only authorised entities.

Their solution focuses on adding encryption mechanism for those control messages. These cryptography mechanisms are given more detail in [18]. In the topology-type attacks, malicious nodes may change the way to process the control messages rather than manipulating them because it will be more difficult to be detected. For example, in the Rank attack, the attackers break the Rank rule rather than modifying any control message. Therefore, it is not enough to use just cryptography to protect the control messages.

The only IDS for protecting the 6LoWPAN-RPL topology attacks, to the best of our knowledge, is the work of Raza *et al.* [79]. Based on that there are several efforts to develop the system, such as the work of Matsunaga *et al*. [100] as an effort to solve the synchronisation issue of the monitoring architecture.

Raza *et al.* [79] proposed SVELTE, an anomaly-based IDS for securing the RPL protocol with two phases including collecting and analysing the IDS data. In the collecting phase, the DODAG root (which is also the monitoring node of all nodes in the network) will request its network members to send information about itself and its neighbours. The information that each member has to send includes the RPL Instance ID, the DODAG ID, the DODAG Version Number, all neighbours and their corresponding ranks, the parent ID, the node's rank, and a timestamp. On receiving the information, the monitoring node starts the analysing phase by using such data to form the network map, evaluate the rank consistency and check the legitimacy of the rank rules between any parent-child pair. Overall, Raza's IDS defines a normal node as a node that has consistent rank value and follow the rank rule. Reflecting on the attacks presented in Section 3.5, this solution does not consider the cases when the attackers generate a lot of additional routing control message to break the stability of network topology (Local repair or DIS attack). In such cases, the malicious nodes still follow the rank rule and have a consistent rank yet downgrade significantly the network performance.

There are a number of works in specifying routing protocols in wireless sensor or ad hoc network to detect internal threats. In such environments, the routing protocols are usually profiled manually by experts through its theoretical specifications. This method is applied in a majority of specification-based proposed solutions [68], however, as involving human expertise, it is lack of flexibility (i. e., depended on the availability of the experts) and hard to verify. For example, there exist several specifications for AODV protocol, but it is difficult to compare or verify their effectiveness. To overcome this problem, the authors in [98] proposed a technique based on Inductive Logic Programming (ILP) method to induce a hypothesis from individual observations and background knowledge. The authors collect examples of the protocol executions through extensive simulation traces and derive an abstract model of protocol behaviour from them. This solution has the advantage of fast profiling generation with the ability of validating the correctness of such specification.

There are also several RPL specifications, but none of them satisfied the purpose of detecting the RPL topology attacks. For example, the specification work in [97], which was only developed for the purpose of conformance testing. In more detail, the authors first tried to specify the behaviours of the host, router, and border router. They then generated test samples, which included optimal sequences of states for each particular node type, to verify if the actual nodes' behaviours follow these sequences. If the

behaviours of the node are as expected, the protocol is concluded to have accurate implementation and vice versa. This RPL specification may not be simply applied for the attack detection purpose for several reasons. First, the testing phase assumes that they will have an actual view of the node behaviours, while the IDS only provides the node behaviours through the view of a monitoring node. The difference is that what the monitoring node sees may not be equal to what actually happens on the monitored node given the synchronisation issue. Second, the sequence of a node's behaviour that the monitoring node collected may not reflect the real sequences because sometimes the information about an activity can be missed. Moreover, the sequence consists of many redundant states and transitions, unlike the optimal test sequence, which requires a more effective way to verify. Third, the RPL specification does not reflect some crucial rules that limit the attackers to attack the optimal topology, such as a node needs to propagate consistent ranks with what are reflected from its neighbours, a node needs to follow the rank rule in any case, or a node needs to not generate redundant control messages.

In the next section, we will develop a specification-based IDS based on the ILP technique while employing the data collection enhanced from [79] to detect the topology-type internal threats.

5.3. Proposed Solution

Our solution consists of two phases. In the first phase, we aim at getting a specification-based model for the IDS. We first simulate the network operation in the normal condition to get the trace file. We then define all the states that relate to the network topology stability and analyse the transitions between those states based on similar algorithms presented in [98]. As discussed in Section 5.2, the approach of using the trace file to generate the operation rules has multiple advantages compared with profiling a protocol from analysing its documents. The generated module can be improved further by expert knowledge added from the insightful understanding of the protocol. We also record the statistic of the states and their transitions, as we know that for some states, the more transitions between them, the more instable the network is. Hence recording the statistic of the important states and setting the threshold for transitions between them will help to detect the topology internal threats more effectively. In the second phase, we translate the knowledge of the specification-based model to the detection algorithm to implement in the IDS server to check the nodes' behaviours. We will use the architecture proposed in the previous chapter to implement the IDS.

5.3.1. Profiling the RPL

To profile the RPL protocol, we used the traces of legitimate protocol behaviours generated from the Contiki-Cooja simulation platform [84]. There are many different behaviours implemented in the protocol, for example, to start a node operation, or to turn the radio cycle on or off. However, as the focus is to protect the optimal topology, we only consider behaviours that related to the optimal and stable topology, in particular, the route establishment and the route maintenance processes. In profiling those behaviours, our specification-based module sets the rules to guarantee that if the internal attackers start to compromise any of the topology operations, they will be detected.

Assume that the network has $n + 1$ sensor node sending the packets to the sink. Such node has ID from $1, \ldots, n$ while ID of the sink is $n+1$. Let CM_k^i be the i^{th} control message collected from node k, which is extracted from the trace file and rearranged following the time order. Let N_k be the total number of control messages collected from node k. We first develop a simple algorithm to extract the states, transitions, and their statistics in each node as below.

Algorithm 1. Extracting states and transitions

Require: Trace file from simulation with marking relevant states

```
1: for k = 1 to n do
2:     for i = 1 to N_k – 1 do
3:         PState = StateExtract(N_i) //Get previous state from N_i
4:         CState = StateExtract(N_i+1) // Get current state from N_i+1
5:         CTran = NewTran(PState, CState) // Get current transition
6:         if CTran ∉ AllTrans[k] do
7:             AllTrans[k] = Add(AllTrans[k], CTran) // Add transition to list
8:             AllTransStatistic[k] = AddStatistic(AllTrans[k], CTran) // Add statistic
9:         else
10:            AllTransStatistic[k] = AddStatistic(AllTrans[k], CTran) // Add statistic
11:        end if
12:    end for
13: end for
```

At the end of Algorithm 1, we generate a set of concrete states, transitions, and corresponding statistic data for each node. For example, we extract the relevant trace to node 3 and observe the following messages: *[Node 3 broadcasts DIS - Node 3 receives DIO from node 5 - Node 3 receives DIO from node 7 - Node 3 receives DIO from node 9 - Node 3 calculates the preferred parent and send a new DIO]*. This trace then will be recorded as *[Node 3 broadcast DIS – Node 3 receives DIOs (3 times) – Node 3 process received DIO – Node 3 send a new DIO]*. Figure 5.1 illustrates an example of the results recorded from Algorithm 1 for Node 3. As can be seen from Figure 5.1, the flow of the

in and out messages to Node 3 is represented in CM[3] on the left, while the transition merge and relevant statistics are represented in AllTrans[3] on the right.

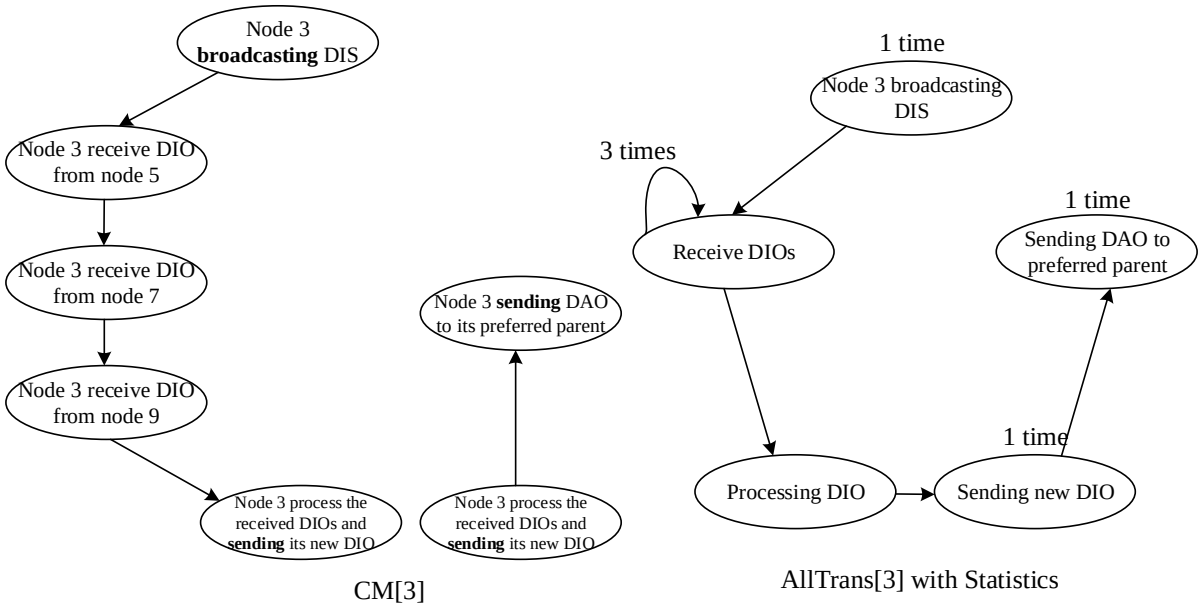


Figure 5.1. Example of the results of Algorithm 1

The results of the Algorithm 1 are sets of states, transitions and corresponding statistic for each node. Algorithm 2 will merge those sets one by one to form an abstract of RPL operation.

Algorithm 2. Form the specification-based IDS for RPL

Require: *AllTrans[k], AllTransStatistic[k], k = 1..n*
1: *FinalSpe = AllTrans[1]*
2: *FinalSpeStatistic = AllTransStatistic[1]*
3: **for** *i = 1* **to** *n* **do**
4: *FinalSpe = Merge(FinalSpe, AllTrans[i])*
5: *FinalSpeStatistic = Merge(FinalSpeStatistic, AllTransStatistic[i])*
6: **end for**

The Merge function first compare the states of two Transitions. It only adds to the FinalSpe the states and transitions that it does not yet have. It also compares the statistic pattern and only records the pattern with significant different trend.

At the end of Algorithm 2, we obtain a Specification-based module as shown in Figure 5.2 below.

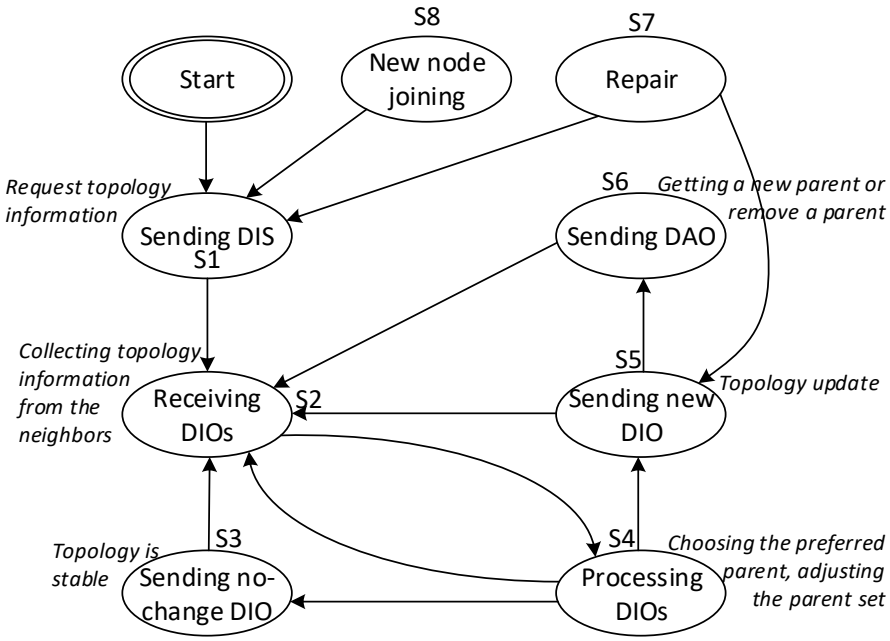


Figure 5.2. Specification-based IDS for RPL through trace file analysis

The generated RPL specification module consists of 8 states: The sending DIS, sending DIO, receiving DIO (DIOr), sending no-change DIO, processing DIOs, sending new (changed) DIO, Sending DAO, repair, and new node joining. From the statistical data recorded, we obtain the following observations:

1. Nodes only move to the ***Sending DIS*** state when it first starts, joins, or involves in a link repair procedure. As a result, nodes only visit the ***Sending DIS*** state a few times during the network performance.
2. Nodes in the centre tend to have more transitions compared to the nodes at the border. The reasons are that centre nodes have more neighbours than border nodes, while their neighbours are also more likely to update the routing information than the border nodes' neighbours are.
3. In the ***processing DIOs*** state, nodes have to follow the rank rule strickly.
4. After a long enough time of running, when the network topology becomes stable, the node will visit mostly the ***Sending no-change DIO*** state. However, such visit is not too often, because the DIO trickle time is always extended in a stable network.
5. The five states ***sending DIS, sending new DIO, sending DAO, repair,*** and ***new node joining*** indicate the instability of the network topology. When the node is in one of these states, the transitions are expected to happen more often, because the DIO trickle time is set to minimum.

In the next section, we will use the knowledge obtained from this section to design and implement a Specification-based IDS for securing the topology attacks toward the RPL network.

5.4. Evaluation Results and Discussions

To investigate further the effectiveness of the IDS, we implement the five types of attacks as discussed in Chapter 3 in Contiki-Cooja [13] and see how the IDS module can detect them. This session first presents the simulation setup and then discuss about the results achieved.

5.4.1. Designing and Implementing the Specification-based for RPL-based Network

5.4.1.1. IDS Design

We employ similar monitoring architecture as discussed in Section 4.2 and 4.4.4 of the previous chapter, in which the monitoring nodes have two interfaces, one is for communicating in 6LoWPAN, and the other is for sending the IDS data to the IDS server. The monitoring nodes will stay as the cluster head to collect the IDS data from its cluster members.

We do not make the cluster head change to promiscuous mode to eavesdrop on all the radio communication around because it will drain its battery out quickly, while the obtained information in this case also has only limited use. Because all the cluster members are neighbours of the cluster head, the head can record all the broadcasting control messages and the unicast message toward itself. For the missing information, we make the cluster head request its members to report periodically. The period time is set up depending on the particular scenario. Once the members receive such request, they need to send their neighbour lists with corresponding ranks, the preferred parent, and its own rank.

In our solution, the procedure to ask the cluster member to send the data to the cluster head will be similar to Raza's work. However, Raza's solution suffers from the synchronisation issue, for example, in case the data from the neighbour is collected later or sooner than data from the node, which make the crosschecking process unreliable. Matsunaga *et al*. [100] pointed out this issue through the following example as can be illustrated in Figure 5.3. Let A is a normal node in the network and N is its neighbour. At

time t_1 when A broadcast its DIO, its rank is 3 and N record rank of A as 3 in its memory. At time t_2, A updates its rank to 4 but because it is not the time to send the new DIO yet, so A does not send any new DIO, and N still store the rank of A as 3. Sooner after t_2, at t_3, the root requests nodes to send IDS information. According to Raza's solution [79], node A will send its rank as 4 (its current rank) while node N informs the root that the actual rank of A is 3. The rank information of node A that it and node N reported is not the same because the recording time was not synchronised. This synchronisation issue makes the root consider that the rank of A is not consistent; hence detect A as malicious node, which create a false detection.

Matsunaga *et al*. [100] proposed an improvement for Raza's solution by letting the nodes send only the rank information in its latest broadcast DIO, rather than the latest rank it has. Moreover, they separate the rank inconsistent threshold when detecting the consistency, in which if there is time difference when receiving the report rank (information from the node itself) and the monitor rank (information from the node's neighbours), the threshold will be higher than it is in the case there is no time difference. Such improvements are claimed to decrease the false detection rate of Raza's solution.

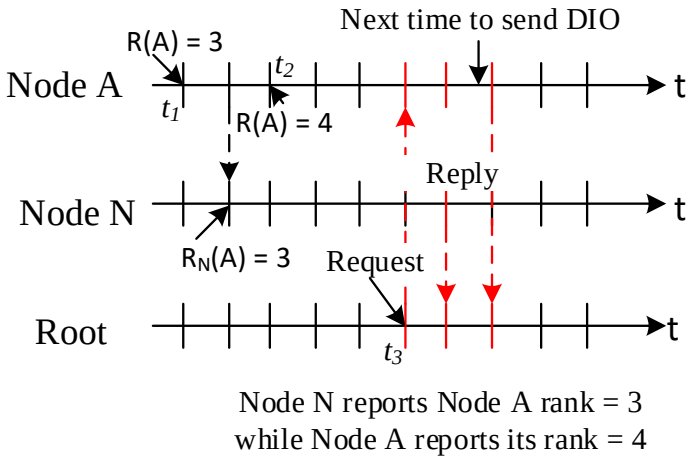


Figure 5.3. Synchronisation issue in Raza's solution

However, Matsunaga's solution still cannot overcome the synchronisation issue. For instance, based on Figure 5.3, we add node P as node A's parent and N is the neighbour of both A and P. At time t_1, P has rank 2, A has rank 3. In time t_1' P updates its rank to 3, hence it broadcasts this new information to the neighbours. At time t_2, node A receives this information and increase its new rank to 4 without updating its rank for the neighbour yet. At time t_3, the root requests every node to send IDS information. According to Matsunaga, P will report its rank as 3 because it has already sent the new DIO before the root request. On the other hand, node A will also report its rank as 3 because its next DIO

is scheduled after the root asked for report. Both of these reports are considered consistent under the view of node N. Now, A and P has the same rank as 3, both are considered consistent, but according to the rank rule, P is the parent of A so it should have a lower rank than A. As a result, both A and P may be considered as malicious source. The illustration for this example is given in Figure 5.4 below.

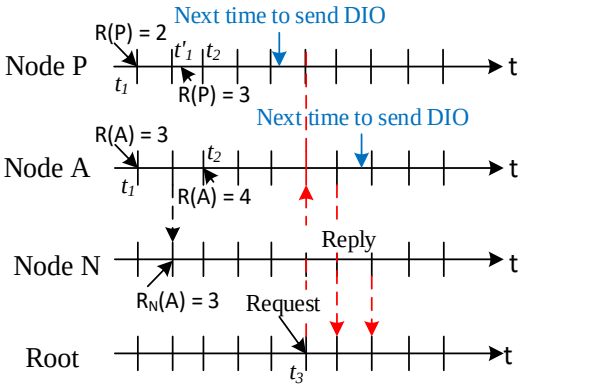


Figure 5.4. Synchronisation issue in Matsunaga's solution

In order to solve this synchronisation issue, we propose to add the sequence number information in the DIO and DIS messages. We will use the reserved bytes in the DIO and DIS message format (readers can refer to [101] for the format of DIO and DIS messages) for this purpose, so the actual size of such messages will remain the same. The synchronisation issue is solved because sequence number indicate specifically which packets the information belongs to; hence, the server can know whether to verify information from two different sources.

5.4.1.2. IDS Implementation

Given the design in Section 4.4.4, the cluster head will record the following information for each of its members.

- DIS sequence, number of DIS received
- DIO sequence, number of DIO received
- List of neighbours, each neighbour has
  - Node ID
  - Rank
  - The sequence of the DIO that provides this info
  - DIS sequence, number of DIS received

- DAO sequence, number of DAO received, and a parent bit (if there is no DAO message sent, or if there is a DAO message require to remove the parent relationship, then the parent bit is 0, otherwise it will be set to 1)

- Preferred parent ID

We develop a detection algorithm based on the Specification-based module in Section 5.3.1 using this provided information to implement in the IDS server. The algorithm consists of 5 modules:

1. M1: Checking the DIS message, alarm if the received DIS is fake or sending too much.
2. M2: Checking the sequence of DIO message, alarm if the received DIO is fake
3. M3: Checking the rank consistency, alarm if the rank of the member is different to the rank reported by its neighbour or the cluster head, given the same DIO sequence. Penalise the neighbours if they do not have the latest DIO message. Alarm if there is any DIO message reported by the neighbours or cluster head that has newer DIO sequence than the member itself
4. M4: Check the rank rule between every pair of parent and child
5. M5: Check the instability of the network area around a member through the relevant states and observations. Penalise if there is any instability and reward if no change happened.

The detail algorithm with these modules is as below.

Algorithm 3. Detecting topology attacks from cluster head view

Module 1: Check whether DIS message is illegitimate

1: **On receiving** *DIS* {
2: **record** *SourceID, DIS_seq_new;*
3: *DIS_count[SourceID]++;*
4: **if** *DIS_seq_new* ≤ *DIS_seq_current* **then**
5: **alarm** *fake DIS;*
6: **else** *DIS_seq_current = DIS_seq_new*
7: **end if**
8: **if** *DIS_count[SourceID]* > $\text{threshold}_{\text{DIS\_count}}$ **then**
9: **alarm** *DIS attack;*
10: **end if** }

Module 2: Check whether there is any fake DIO

1: **On receiving** *DIO* {
2: **record** *SourceID, DIO_seq_new, rank;*
3: *DIO_count[SourceID]++;*
4: **if** *DIO_seq_new* ≤ *DIO_seq_current* **then**

```
5:      alarm fake  DIO;
6:      else DIO_seq_current = DIO_seq_new
7:   end if   }
```

Module 3: Check the rank inconsistency

```
1:   After receiving reports from all of the members {
2:      for each Member in Cluster do {
3:        if Member.DIO_seq < CH.Member.DIO_seq then
4:            alarm fake DIO;
5:        end if
6:        for each Neighbour in Member.Neighbour do
7:            if Member.DIO_seq < Neighbour.Member.DIO_seq then
8:                alarm fake DIO;
9:            else if Member.DIO_seq < Neighbour.Member.DIO_seq then
10:                Neighbour.fault = Neighbour.fault + 0.5 //penalised
11:           else if Member.DIO_seq == Neighbour.Member.DIO_seq then
12:               if Member.rank != Neighbour.Member.Rank then
13:                   alarm fake DIO;
14:               end if
15:           end if
16:       end for
17:     end for } }
```

Module 4: Check the rank rule

```
1:  for each Member in Cluster do
2:      if Member.rank + MinHopRankIncrease < Member.parent.rank then
3:          alarm rank attack;
4:      end if
5:      for each Neighbour in Member.Neighbour do {
6:          if Member.DAO.parent == 1 then
7:              if Member.rank - MinHopRankIncrease > Member.child.rank  then
8:                  alarm rank attack;
9:              end if
10:         end if
11:   end for
12: end for }
```

Module 5: Check the stability of the network part which relate to a cluster member

```
//Setting the initial stability evaluation for each member in cluster
1: for each Member in Cluster do
2:      Member.stability = threshold_stability
3: end for
//Penalise if stability condition is observed to be not satisfied
4: for each Member in Cluster {
5:      if IsRepairAfterPeriod then
6:          Member.stability -= 2 //penalised -2 on stability
7:      end if
8:      if IsChangeAfterPeriod(Member.DIO) || IsChangeAfterPeriod(Member.DAO) ||
            IsNewNodeJoiningAfterPeriod then
9:          Member.stability -= 0.5 // penalised 0.5 on stability
10:     end if }
//Checking every period of time
11: if IsCheckingPeriod then
12:     for each Member in Cluster do
13:         if Member.stability < 0 then
```

14: **alarm** *Member instability*;
15: **end if**
16: **if** *Member.fault* > $threshold_{fault}$ **then**
17: **alarm** *Member fault;//* member fault is recorded in module 3
18: **end if**
19: **end for**
20: **end if**

The thresholds used in the algorithm are summarised in Table 5.1 below.

Table 5.1. The thresholds used in the algorithm

Threshold	Meaning
$threshold_{DIS\_count}$	Alarm if a node visit the DIS state more than $threshold_{DIS\_count}$ in the monitoring time.
$threshold_{fault}$	Alarm if a node not updating info from the neighbours
$threshold_{instability}$	Alarm if node visit the instability states S1, S5, S6, S7, S8 more than a $threshold_{instability}$ in the monitoring time

5.4.2. Simulation Set up

We reuse the topology set up in Chapter 4, as presented in Section 4.5.1 and Figures 4.11 and 4.12. To recall, our simulation scenario consists of 100 nodes placed randomly in a 600x600m area, each node has a transmission range of 50m. There is one sink placed in the top left and 11 IDS cluster heads to cover the operation of the remaining 88 nodes in the network. Every node sends packet to the sink at the rate of 1 packet every 60 seconds. We implement the specification-based module from Section 5.3 in the IDS server. The cluster heads are chosen manually so that they can cover the monitoring of all the nodes in the network. Cluster heads collect the IDS data from cluster members in the form shown in Section 5.3 before sending to the IDS server. The period time that the cluster head requests its members to report the IDS information is 2 minutes. The thresholds are set up with the following values: $threshold_{DIS\_count} = 3$; $threshold_{fault} = 2$; and $threshold_{instability} = 10$. Note that $threshold_{DIS\_count} = 3$ *and* $threshold_{fault} = 2$ are the two fixed values to detect the anomaly in breaking the RPL operation. This is because in a normal network, a normal node would not normally exceed these two thresholds. On the other hand, the $threshold_{instability} = 10$ is set due to analysing the statistical data in the trace files, which is applicable for our particular scenarios. In the other scenario, this threshold can be changed accordingly. If the operators wants to have a stable network most of the time, this value can be decreased. If they feel that the environment may create unstable topology, they can increase this value to prevent the increase of IDS sensitivity.

We implemented each of the four types of attacks in Section 3.5, including Sinkhole attack, Rank attack, Local Repair, and DIS attack, in a random position of 88 normal senders. The attacks are initiated after 3 minutes when the topology establishment phase

is done. As the implementations of multiple interface sensors not yet available in Contiki, we run the detection process through analysing the simulation trace files as the proof-of-concept. The summary of the simulation parameters is shown in Table 5.2 below.

Table 5.2. The simulation parameters

Parameters	Value
Simulation platform	Cooja Contiki 2.6
Number of nodes	99 senders, 1 sink
Number of cluster head	11
Number of attackers	1
Traffic model	Constant bit rate
Sending rate	1 packet every 60 second
IDS require info every	2 minutes
Simulation run time	31.5 minutes

5.4.3. Simulation Results and Discussions

5.4.3.1. Detection Efficiency

We divided the RPL attacks discussed in Section 3.5.2 into two groups, which have similar results when detected by our IDS. The first group contains the Rank attack, Sinkhole attack - the threats which are detected only by the specified states. The second group includes the Local Repair and the DIS attacks, which involved both the specification states and statistic collection to reveal.

Table 5.3 below shows the TPR and FPR of the Rank attack (RA), Sinkhole attack (SA) after 4 minutes, when the Rank attack already initiated (at minute 2) and the IDS has just collected the first two IDS data packets from its neighbour. As can be seen from the table, we obtained ideal IDS results, where the TPR is 100% and the FPR is 0%. These results can be explained as follows.

Table 5.3. TPR and FPR of Rank attack, sinkhole attack and Neighbour attack after 4 minutes

	TPR(%)	FPR(%)
RA	100	0
SA	100	0

The DIO sequence checking in Module 2 and 3 ensure that there is no inconsistency in DIO and Rank information between the IDS data reported by the neighbours. As a result, the cluster head will know most of the parent-child relationships and their accurate corresponding ranks.

Regarding the Rank attack, given its nature, which is choosing the worst parent as the preferred parent and changing it frequently, the cluster head will detect the behaviour of

breaking the rank rule and raise an alarm about the child node for choosing inappropriate parent. Therefore, the Rank attack is detected quickly with high accuracy.

Regarding the Sinkhole attack, our implementation lets the attacker keeps informing that it has the rank of the Sink to attract its neighbours. As the attacker is not the actual Sink, before initiating the attacks, it would have a preferred parent. Moreover, this parent-child relationship would be recorded by one of the cluster heads. When the attacker manipulates its new rank to the Sink's rank, such relationship will become illegal, because the child now has a better rank than the parent does. This illegal relation will be detected by Module 4 of our IDS.

Our IDS shows high accuracy results not long after the attack initiating. However, when letting the IDS works for a long time, when the TPR is still ideal, the FPR increases significantly and makes the IDS become less accurate. For example, the TPR and FPR after 10 minutes detecting RA, SA and NA scenarios are shown in Table 5.4 below. The table shows that the FPR increase to about 2-5%. The reason is that the initiated attacks in the tampered nodes have affected its neighbours around, make those nodes work the same way as the attackers, and therefore become difficult to separate.

Table 5.4. TPR and FPR of Rank attack, sinkhole attack and Neighbour attack after 10 minutes

	TPR(%)	**FPR(%)**
RA	100	5.25
SA	100	3.28

In order to minimise the FPR, the nodes which are detected as the malicious source should be removed from the network, for example, by adding to a blacklist and asking all other relevant nodes to skip nodes in that list. After removing the nodes, the IDS will stop judging for a certain time to help to stabilise the network before restarting in a new detection cycle.

The second group of attacks includes the Local repair and the DIS attack. The difference between this group and the first group is that in this group, observing that a node visiting a state is not enough to conclude that this node is a malicious node. This observation is considered only as part of the statistical evidence. Only when a node visits a state more than a threshold of times during a period, the IDS has the right to raise alarm about the threat.

The mechanisms to detect attacks in the second group are as follows. In Local repair attack, after initiating the local repair mechanism, the node sends the poison messages to

the neighbourhood in which its rank is reset to be infinite and it needs to resend the DIS to obtain the surrounding routing information. The local repair will be reported after several times initiated according to Module 1. On the other hand, the local repair also invokes the high instability value in Module 5, so it will also be reported by this module.

In the DIS attack, the attacker needs to send DIS messages to force the neighbour to change the DIO trickle time, or to send the unicast DIO back. In both cases, it will increase the DIS statistic in Module 1; and this will be reported by the IDS.

Table 5.5 and 5.6 below present the TPR and FPR detection of this group after 8 and 12 minutes respectively. As can be seen from the tables, after 8 minutes, the IDS may not collect enough information in any of the cases so it cannot detect the Local repair attack and DIS attack, which results in a high FN and low TPR. On the other hand, after 12 minutes, the IDS collect all the needed information, so the FN and TPR are ideal. However, the Local repair and DIS attack is given long time enough to manipulate the neighbours around the malicious node to create the instability in the topology. Such instability is presented through the high rate of FP and FPR, which make the IDS, become less accurate because of detecting normal nodes as attackers. Therefore, there is a trade-off between the TPR and FPR in detecting threats in this group.

Table 5.5. TPR and FPR of Rank attack, sinkhole attack and Neighbour attack after 8 minutes

	TPR(%)	**FPR(%)**
LA	86.36	0.67
DIS	94.32	3.03

Table 5.6. TPR and FPR of Rank attack, sinkhole attack and Neighbour attack after 12 minutes

	TPR(%)	**FPR(%)**
LA	100	6.78
DIS	100	5.92

Unlike the first group of attacks, where the attackers are always detected first before any False Positive happens, in the second group, a benign node can be detected as malicious before the attackers are revealed. Therefore, using a blacklist or other mechanism to eliminate the detected nodes in this case may not be appropriate. An alternative solution is to adjust the frequency threshold to adapt to the frequency of the corresponding Local repair/DIS attack in specific scenario through learning from simulation. A good chosen threshold will optimise the detection rate and accurateness of the IDS.

Our IDS aims at monitoring the optimal network topology and its stability, which is something broader than just monitoring the node performance. Hence, it will also have

the ability to detect other topology attacks that are not mentioned in this chapter, if those attacks break the optimal topology and create network instability.

5.4.3.2. Energy Efficiency

Similar to Section 4.5.3.2, we will not consider the overhead in communicating between the cluster heads and the IDS server, and the computing waste at the server side.

From the 6LoWPAN side, we run the simulation in RPL-collect network with and without our IDS integration. We obtain the energy and power consumption as calculated in formula (1) and (2) in Section 2.6.5. In normal RPL network, the energy is around 190J, while in the RPL with IDS integration, the average energy consumption is about 202J, which represents an increase of 6.3%, which can be considered small level of overhead. The average power consumption in each node increases slightly 6.3% with 1.2mW in IDS integration scenario to compare with 1.05mW in normal scenario. This indicates that the network lifetime will not be affected much once implementing the IDS.

With our architecture design, adding more nodes in IDS-cluster forms will not make nodes in the network consume more energy, therefore we can extend the scale of the network easily. The overhead of the added IDS for the adding nodes will be local and not affect to the previous setup of the network. Hence, we can conclude that the cluster monitoring structure is resource efficient and can expand well to the large-scale network.

5.5. Chapter Summary

This chapter looked at detecting the topology attacks towards RPL performance, particularly in breaking its optimal topology and creating the instability. The considered attacks include the Rank, the Sinkhole, the Local Repair, and DIS attack. Our detection solution involves semi-auto building a specification-based IDS model for protecting RPL-based network topology. The main idea is to learn the states, transitions, and relevant statistics based on the analysing the trace file. The generated model will be integrated in the IDS server, which receives IDS data from cluster heads through a clustering monitoring architecture. The simulation results show that our IDS is energy efficiency and scalable, while providing high detection rates and accurateness in revealing most of the topology attacks.

The combination of the specification-based IDS in this chapter and the Bayesian-based IDS in Chapter 4 will form a robust IDS module, which can detect most of the potential

internal threats to the 6LoWPAN network. Such a combination is feasible given the IDS architecture designed in this thesis, because there are not overhead created when combining the two IDS modules while the computation and storage workload are put at the IDS server side, which has extensive capability due to the use of cloud computing technology. In the next chapter, we will conclude the thesis and give the future plan.

CHAPTER 6. CONCLUSION AND FUTURE WORK

This chapter will summarise all the research efforts presented in the previous chapters together with their overall achievement to the aim and objective of the research. After that, all possible modifications, which could improve the performance of the presented research solutions, will be discussed in detail as future work.

6.1. Contribution of the Research

This thesis has focused on the design, implementation, and evaluation of IDS solutions for detecting the internal attackers in 6LoWPAN network.

First, we have categorised the internal threats into two main types: the performance-type and the topology-type. The common objectives of the former are the traffic towards the neighbours, the channel around the malicious nodes, and the creating of the fake neighbours to confuse the communication of other nodes. On the other hand, the latter particularly focuses on manipulating some RPL properties, which are designed specifically for the optimizing purpose, for instance, the Rank for optimizing the connection between the parent node and the child, the repair mechanisms for route maintenance, or the DIO trickle time for reducing the overhead. We have proposed different novel ways of manipulating those properties to downgrade the network performance significantly, like presented in the Rank attack, the Local Repair and the DIS attack. We have also studied the typical threats in each type, which are the Black Hole, Grey Hole, Delaying, Jamming or Hello flood, in the performance-type; and the Sinkhole, Rank, Local Repair, and DIS attack in the topology-type. For each of the attack, we studied the general impact to the network performance through the delivery rate, delay, and control overhead. We also identified the sensitive metrics that changed significantly when there is an abnormal behaviour. We later employed these metrics as the evidences to determine the occurrence of the internal attackers.

Second, for detecting the performance-type attack, we develop a BN as a statistical tool to monitor multiple metrics. The development process consists of four phases, from identifying the BN nodes and their dataset; supervised and unsupervised learning the structure of the nodes; learning the parametric between the nodes; to implement this model into the network. Our Bayesian module uses the simulation trace data from normal as well as different attacked scenarios for training. Such training data provide the

thresholds for differentiating the nodes' states, the potential cause-effect relations between the nodes, while also serves as the input for parametric learning. To compare with other solutions, our Bayesian-based IDS module can take into account simultaneously more input metrics, which will enable a deeper understanding of the behaviour. The module can also record the history of nodes' performance by using a counter of near-abnormal state, which allows to judge the node long-term behaviours. For example, when a node has continuously bad performance, but still out of the abnormal range, our IDS can turn the judgement to the abnormal state using this counter bit, which helps to detect attackers which aiming at long-term through lowering their short-term impact to bypass the IDS. The simulation experiments as presented in Chapter 4 shows that our module achieved a detection rate from 87% to 100%, while mostly having a false positive rate under 5%, which is an encouraging result for IDS approaches. Moreover, the estimated overhead created by this module is insignificant, which allows the implementation in real scenarios.

Third, for detecting the topology-type attack, we develop a specification-based IDS for RPL. By comparing to the main properties in the RPL profile, we can judge the nodes' behaviours to see if they follow a legitimate pattern or not. We specified the RPL through studying both the simulation traces of the normal and attacked scenarios, as well as employing the expert understanding on the protocol manual. The result of this studying is a model of relevant RPL states, transitions, and statistic based, which were utilised further in implementing the Specification-based IDS module in 6LoWPAN. In implementation, we also proposed to add the sequence number property in the spare field in the RPL DIO and DIS control messages, to solve the synchronisation issue of the previous RPL data collection solution.

For both of the Bayesian-based and Specification-based, we use a clustering architecture that allows the cluster heads to collect the IDS data from its cluster members before sending such data to the IDS server. We propose to provide cluster heads with an additional interface to communicate effectively with the server side, while the IDS server can use the cloud computing to expand its capability. The simulation results show that this monitoring architecture is energy efficient and highly scalable.

Each of IDS modules is designed for detecting a specific type of internal attacks. The use of a single module, therefore, will not be effective to deal with all the internal threats. For example, the BN module cannot effectively detect the topology attack because such

attacks create similar behaviour patterns between the attackers and their neighbours, which will create either high false alarm rate or unable to detect the attackers. On the other hand, the use of the RPL Specification-based module only cannot detect effectively the performance-type attacks because most of these attackers satisfy the legitimate RPL behaviours. With the IDS architecture proposed in this thesis, 6LoWPAN can run both the Bayesian-based and Specification-based simultaneously without much overhead created, because the cluster heads only do the recording, pre-processing, and sending the IDS data to the IDS server. The main computing and storing workload are at the IDS server, which has extensive capability due to applying of cloud computing technology. As a result, the combined system will be able to deal with most of the typical internal attacks. However, the combination of these two IDSs can still be optimised further. For example, to reduce the redundancy in the monitoring information that the cluster members reported to the cluster heads, or to combine the algorithms to further decrease the computing workload. Due to the limited time and resource, such work will be left as future work.

In our view, a 6LoWPAN can only be guaranteed to be free from the internal threats when and only when (i) every node works with its optimal capability, including having optimal traffic, strong communication channel, and all the nodes have legitimate neighbours; and (ii) nodes in the network always operate follow optimal topology and protocol rules. By designing and building the two aforementioned IDS modules aiming directly at these two objectives, we have achieved the IDSs, which have the ability to detect most kind of the typical internal threats, which is an important contribution to the security of 6LoWPAN.

6.2. Future Work

This section presented some future potential work, which can follow on from the research in this thesis, to strengthen the proposed solutions as well as open new research directions:

- The last chapters have proposed the two different IDSs, the Bayesian-based and the Specification-based modules to detect two different types of the internal threats. These two modules now operate independently in 6LoWPAN; however, a combination of them may optimise the monitoring procedure as well as reduce the overhead. Due to the limitations of time and resource, such combinations could not be developed properly in this thesis. Potential combination directions can be integrating the Specification-based approach as part of the Bayesian-based

approach. In this way, the properties to monitor of the Specification-based should be re-optimised and assessing the relations with other properties in the Bayesian-based, to provide more closed relations. These relations may improve the accurateness or reduce the system's overhead accordingly.

- The BN module has the strong ability in statistically answering different problem with the network performance. Hence, the module can be extended to other issues like routing optimisation, or Medium Access Control. The use of BN with different metrics in different layers may create a promising cross layer design that achieves better performance results.
- The protocol profiling techniques in Chapter 5 can be widely applied to many other routing protocols in different networks for similar security purpose.
- Recently, technology development regarding studying Big data, Hadoop platform, and cloud computing has promised to bring the Internet application to a new level. The application of these technologies provides faster processing yet larger data consideration, which will significantly improve the accurateness and effectiveness of our solution. Hence, our future work is also to implement our systems using these technologies to extend their capability.
- Due to the limitations of time and resource, our research focused only on a single attacker using single attack to downgrade the network. In reality, attackers can make their attack much more complex to inflict more damage. For example, malicious sources can apply different attacking mechanisms in a specific order to increase the impact yet change the behavioural patterns so that the detecting system cannot detect them. On the other hand, adversaries can initiate the attack simultaneously in several positions of the network. Attack sources in this case can cooperate to maximize the damage or bypass the detection system. We will extend our IDS to deal with these multiple attack issues in the future.

Reference

[1] Y. Guang-xue and L. Feng-jiao, "Investigation of Security and Defense System for Home Based on Internet of Things," in *Web Information Systems and Mining (WISM), 2010 International Conference on*, 2010, pp. 8-12.

[2] S. Guicheng and L. Bingwu, "The visions, technologies, applications and security issues of Internet of Things," in *E -Business and E -Government (ICEE), 2011 International Conference on*, 2011, pp. 1-4.

[3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, pp. 2787-2805, 2010.

[4] ITU. (2005). *Internet Reports 2005: The Internet of Things* Available: http://www.itu.int/osg/spu/publications/internetofthings/

[5] X. Li, Z. Xuan, and L. Wen, "Research on the architecture of trusted security system based on the Internet of things," in *Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on*, 2011, pp. 1172-1175.

[6] C. Hartung, J. Balasalle, and R. Han, "Node compromise in sensor networks: The need for secure systems," *Department of Computer Science University of Colorado at Boulder*, 2005.

[7] H. Kim, "Protection against packet fragmentation attacks at 6lowpan adaptation layer," in *Convergence and Hybrid Information Technology, 2008. ICHIT'08. International Conference on*, 2008, pp. 796-801.

[8] A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo, "6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach," *International Journal of Communication Systems*, vol. 25, pp. 1189-1212, 2012.

[9] A. Le, J. Loo, A. Lasebae, A. Vinel, C. Yue, and M. Chai, "The Impact of Rank Attack on Network Topology of Routing Protocol for Low-Power and Lossy Networks," *Sensors Journal, IEEE*, vol. 13, pp. 3685-3692, 2013.

[10] A. Le, J. Loo, L. Yuan, and 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*, 2013, pp. 000789-000794.

[11] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*: Wiley-Blackwell 2009.

[12] J. Martocci, P. D. Mil, N. Riou, and W. Vermeylen, "RFC 5867: Building Automation Routing Requirements in Low-Power and Lossy Networks: http://tools.ietf.org/html/rfc5867," 2010.

[13] A. Brandt, J. Buron, and G. Porcu, "RFC 5826: Home Automation Routing Requirements in Low-Power and Lossy Networks http://tools.ietf.org/html/rfc5826," 2010.

[14] K. Pister, P. Thubert, S. Dwars, and T. Phinney, "RFC 5673: Industrial Routing Requirements in Low-Power and Lossy Networks http://tools.ietf.org/html/rfc5673," 2009.

[15] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, "RFC 5548: Routing Requirements for Urban Low-Power and Lossy Networks http://tools.ietf.org/html/rfc5548," 2009.

[16] J.-P. Vasseur and A. Dunkels, *Interconnecting Smart Objects with IP: The Next Internet*: Morgan Kaufmann, 2010.

[17] L. M. L. Oliveira, A. F. de Sousa, and J. J. P. C. Rodrigues, "Routing and mobility approaches in IPv6 over LoWPAN mesh networks," *International Journal of Communication Systems,* vol. 24, pp. 1445-1466, 2011.

[18] T. Winter, P. Thubert, A. Brandt, T. Clausen, J. Hui, R. Kelsey, *et al.* (2011, RPL: IPv6 Routing Protocol for Low power and Lossy Networks - draft-ietf-roll-rpl-19: http://tools.ietf.org/html/draft-ietf-roll-rpl-19.

[19] N. Kushalnagar, G. Montenegro, and C. Schumacher, "RFC 4919: IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals http://tools.ietf.org/html/rfc4919," August 2007.

[20] R. Roman and J. Lopez, "Integrating wireless sensor networks and the Internet: a security analysis," presented at the Internet Research 2009.

[21] E. J. Cho, J. H. Kim, and C. S. Hong, "Attack model and detection scheme for Botnet on 6LoWPAN," presented at the Proceedings of the 12th Asia-Pacific network operations and management conference on Management enabling the future internet for changing business and new computing services, Jeju, South Korea, 2009.

[22] T. Kavitha and D. Sridharan, "Security Vulnerabilities In Wireless Sensor Networks: A Survey " *Journal of Information Assurance and Security 5,* 2009.

[23] J. Granjal, E. Monteiro, and J. S. Silva, "Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey," *Ad Hoc Networks,* vol. 24, pp. 264-287, 2015.

[24] E. Shi and A. Perrig, "Designing secure sensor networks," *Wireless Communications, IEEE* vol. 11, pp. 38 - 43 2004

[25] B. Kannhavong, H. Nakayama, N. Kato, A. Jamalipour, and Y. Nemoto, "A study of a routing attack in OLSR-based mobile ad hoc networks," *International Journal of Communication Systems,* vol. 20, pp. 1245-1261, 2007.

[26] P. Nikander, J. Kempf, and E. Nordmark, "RFC 3756: IPv6 Neighbor Discovery (ND) Trust Models and Threats http://www.ietf.org/rfc/rfc3756.txt," 2004.

[27] A. Mayzaud, R. Badonnel, and I. Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things," *International Journal of Network Security,* 2016.

[28] A. Dvir, T. Holczer, and L. Buttyan, "VeRA-version number and rank authentication in rpl," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*, 2011, pp. 709-714.

[29] M. Landsmann, M. Wahlisch, and T. Schmidt, "Topology Authentication in RPL," in *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, 2013, pp. 73-74.

[30] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "A study of rpl dodag version attacks," in *Monitoring and Securing Virtualized Networks and Services*, ed: Springer, 2014, pp. 92-104.

[31] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*, 2008, pp. 245-256.

[32] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab, "NanoECC: Testing the limits of elliptic curve cryptography in sensor networks," in *Wireless sensor networks*, ed: Springer, 2008, pp. 305-320.

[33] J. Ayuso, L. Marin, A. J. Jara, and A. F. G. Skarmeta, "Optimization of Public Key Cryptography (RSA and ECC) for 8-bits Devices based on 6LoWPAN," presented at the 1st International Workshop on the Security of the Internet of Things (SecIoT'10), 2010.

[34] Z. Liu, J. Groszschaedl, and I. Kizhvatov, "Efficient and Side-Channel Resistant RSA Implementation for 8-bit AVR Microcontrollers," presented at the 1st International Workshop on the Security of the Internet of Things (SecIoT'10), 2010.

[35] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security (TISSEC),* vol. 8, pp. 41-77, 2005.

[36] T. Chung and U. Roedig, "DHB-KEY: an efficient key distribution scheme for wireless sensor networks," in *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, 2008, pp. 840-846.

[37] S. Raza, S. Duquennoy, T. Chung, T. Voigt, and U. Roedig, "Securing communication in 6LoWPAN with compressed IPsec," in *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, 2011, pp. 1-8.

[38] J. Granjal, R. Silva, E. Monteiro, J. Sa Silva, and F. Boavida, "Why is IPSec a viable option for wireless sensor networks," in *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, 2008, pp. 802-807.

[39] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 packets over IEEE 802.15. 4 networks," No. RFC 4944, 2007.

[40] R. Riaz, K.-H. Kim, and H. F. Ahmed, "Security analysis survey and framework design for ip connected lowpans," in *Autonomous Decentralized Systems, 2009. ISADS'09. International Symposium on*, 2009, pp. 1-6.

[41] Y. Wang, *Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection*: IGI Global, 2008.

[42] Y. Mao, "A semantic-based intrusion detection framework for wireless sensor network," presented at the Networked Computing (INC), 2010 6th International Conference 2010.

[43] R.-C. Chen, Y.-F. Haung, and C.-F. Hsieh, "Ranger Intrusion Detection System for Wireless Sensor Network with Sybil Attack based on Ontology," in *third WSEAS Int. Conf. on BIOMEDICAL ELECTRONICS and BIOMEDICAL INFORMATICS (BEBI '10),* , 2010.

[44] S. J. Lee, H. Y. Lee, and T. H. Cho, "A Threshold Determining Method for the Dynamic Filtering in Wireless Sensor Networks Based on Fuzzy Logic," *IJCSNS International Journal of Computer Science and Network Security,* vol. 8, April 2008 2008.

[45] S. H. Chi and T. H. Cho, "Fuzzy Logic Anomaly Detection Scheme for Directed Diffusion Based Sensor Networks," presented at the FSKD 2006, 2006.

[46] B. Parekh and H. Cam, "Minimizing False Alarms on Intrusion Detection for Wireless Sensor Networks in Realistic Environments," presented at the Military Communications Conference, 2007.

[47] R. Dong, L. Liu, J. Liu, and X. Xu, "Intrusion Detection System Based on Payoff Matrix for Wireless Sensor Networks," presented at the Genetic and Evolutionary Computing, 2009.

[48] M. Estiri and A. Khademzadeh, "A game-theoretical model for intrusion detection in wireless sensor networks," presented at the Electrical and Computer Engineering (CCECE), 2010 23rd Canadian Conference, Calgary, AB 2010

[49] M. Estiri and A. Khademzadeh, "A theoretical signaling game model for intrusion detection in wireless sensor networks," presented at the Telecommunications Network Strategy and Planning Symposium (NETWORKS), 2010 14th International, Warsaw 2010.

[50] S. Banerjee, C. Grosan, A. Abraham, and P. K. Mahanti, "Intrusion detection on sensor networks using emotional ants," *International Journal of Applied Science and Computations,* vol. 12, pp. 152-173, 2005.

[51] E. Soroush, J. Habibi, and M. S. Abadeh, "Intrusion Detection Using a Boosting Ant Colony Based Data Miner," presented at the Proceedings of the 11th International CSI Computer Conference, 2006.

[52] W. Xiong and C. Wang, "Feature Selection: A Hybrid Approach Based on Selfadaptive Ant Colony and Support Vector Machine," presented at the International Conference on Computer Science and Software Engineering, 2008.

[53] S. Kaplantzis, A. Shilton, N. Mani, and Y. A. Sekercioglu, "Detecting Selective Forwarding Attacks in Wireless Sensor Networks using Support Vector Machines," presented at the Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference, Melbourne, Qld. , 2007.

[54] T. J. Wilson, "MFIRE-2: A Multi Agent System for Flow-Based Intrusion Detection Using Stochastic Search," DTIC Document2012.

[55] R. Zhang, D. Qian, C. Ba, W. Wu, and X. Guo, "Multi-agent based intrusion detection architecture," in *Computer Networks and Mobile Computing, 2001. Proceedings. 2001 International Conference on*, 2001, pp. 494-501.

[56] H. Qi, Y. Xu, and X. Wang, "Mobile-agent-based collaborative signal and information processing in sensor networks," *Proceedings of the IEEE,* vol. 91, pp. 1172-1183, 2003.

[57] S. Hamedheidari and R. Rafeh, "A novel agent-based approach to detect sinkhole attacks in wireless sensor networks," *Computers & Security,* vol. 37, pp. 1-14, 2013.

[58] T. V. Phuong, L. X. Hung, S. J. Cho, Y.-K. Lee, and S. Lee, *An Anomaly Detection Algorithm for Detecting Attacks in Wireless Sensor Networks* vol. Lecture Notes in Computer Science, 2006, Volume 3975/2006: Springer, 2006.

[59] Y. Ponomarchuk and D.-W. Seo, "Intrusion detection based on traffic analysis in wireless sensor networks," presented at the Wireless and Optical Communications Conference (WOCC), 2010 19th Annual, Shanghai 2010

[60] X. Song, G. Chen, and X. Li, "A Weak Hidden Markov Model based intrusion detection method for wireless sensor networks," presented at the Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference, Guilin 2010

[61] S. H. Chen and C. A. Pollino, "Good practice in Bayesian network modelling," *Environmental Modelling & Software,* vol. 37, pp. 134-145, 2012.

[62] M. Momani, "Bayesian Fusion Algorithm for Inferring Trust in Wireless Sensor Networks," *Journal of Network,* vol. 5, July 2010.

[63] B. M. David and R. T. d. S. Jr, "A Bayesian Trust Model for the MAC Layer in IEEE 802.15.4 Networks," presented at the I2TS 2010 - 9th International Information and Telecommunication Technologies Symposium, 2010.

[64] T. Koski and J. Noble, *Bayesian networks: an introduction* vol. 924: John Wiley & Sons, 2011.

[65] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, *et al.*, "Specification-based anomaly detection: a new approach for detecting network intrusions," presented at the Proceedings of the 9th ACM conference on Computer and communications security, Washington, DC, USA, 2002.

[66] N. Stakhanova, S. Basu, and J. Wong, "On the symbiosis of specification-based and anomaly-based detection," presented at the Computers & security 29 (2010), 2010.

[67] P. Ning and K. Sun, "How to Misuse AODV: A Case Study of Insider Attacks against Mobile Ad-Hoc Routing Protocols," presented at the Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society 2003.

[68] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for AODV," in *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, 2003, pp. 125-134.

[69] J. Grönkvist, A. Hansson, and M. Sköld, "Evaluation of a Specification-Based Intrusion Detection System for AODV," presented at the The Sixth Annual Mediterranean Ad Hoc Networking WorkShop, Corfu, Greece, 2007.

[70] C. H. Tseng, T. Song, P. Balasubramanyam, C. Ko, and K. Levitt, "A Specification-Based Intrusion Detection Model for OLSR," presented at the Recent Advance in Intrusion Detection RAID 2005, 2005.

[71] J.-M. Orset, B. Alcalde, and A. Cavalli, "An EFSM-Based Intrusion Detection System for Ad Hoc Networks," in *Lecture Notes in Computer Science, Volume 3707/2005*, Springer, Ed., ed, 2005, pp. 400-413.

[72] L. Mostarda and A. Navarra, "Distributed Intrusion Detection Systems for Enhancing Security in Mobile Wireless Sensor Networks," *International Journal of Distributed Sensor Networks,* vol. 4, pp. 83–109, 2008.

[73] H. W. Hesiri Weerasinghe and H. F. Huirong Fu, "Preventing cooperative black hole attacks in mobile ad hoc networks: Simulation implementation and evaluation," *International Journal of Software Engineering and Its Applications (IJSEIA),* vol. 2, pp. 39-54, 2008.

[74] M. Riecker, D. Thies, and M. Hollick, "Lightweight detection of denial-of-service attacks in practical wireless sensor networks: A systematic approach," Technical report, Technische Universität Darmstadt, http://www. seemoo. de/dl/seemoo/seemoo-tr-2014-01. pdf2014.

[75] A. P. R. da Silva, M. H. Martins, B. P. Rocha, A. A. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, 2005, pp. 16-23.

[76] A. Strikos, "A full approach for intrusion detection in wireless sensor networks,"." School of Information and Communication Technology, KTH.2007.

[77] F. Amini, V. B. Miˇsi´c, and J. Miˇsi´c, "Chapter 6. Intrusion Detection in Wireless Sensor Networks," in *Security in distributed, grid, mobile, and pervasive computing*, Y. Xiao, Ed., ed: Auerbach Publication, 2007.

[78] X. Wang and H. Qian, "Hierarchical and low-power IPv6 address configuration for wireless sensor networks," *International Journal of Communication Systems,* pp. n/a-n/a, 2011.

[79] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad hoc networks*, vol. 11, pp. 2661-2674, 2013.

[80] T. H. Hai and E.-N. Huh, "Detecting selective forwarding attacks in wireless sensor networks using two-hops neighbor knowledge," in *Network Computing and Applications, 2008. NCA'08. Seventh IEEE International Symposium on*, 2008, pp. 325-331.

[81] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *Network, IEEE,* vol. 20, pp. 41-47, 2006.

[82] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, pp. 46-57.

[83] A. Le, J. Loo, L. Yuan, and A. Lasebae, "Specification-based IDS for securing RPL from topology attacks," in *Wireless Days (WD), 2011 IFIP*, 2011, pp. 1-3.

[84] Contiki. *Contiki. Cooja Contiki. Available: http://www.contiki-os.org/*.

[85] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-based Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[86] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, pp. 293-315, 2003.

[87] Z. A. Baig, "Pattern recognition for detecting distributed node exhaustion attacks in wireless sensor networks," *Computer Communications*, vol. 34, pp. 468-484, 2011.

[88] Q. Ren and Q. Liang, "Secure media access control (MAC) in wireless sensor networks: Intrusion detections and countermeasures," in *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, 2004, pp. 3025-3029.

[89] J. R. Douceur, "The sybil attack," in *Peer-to-peer Systems*, ed: Springer, 2002, pp. 251-260.

[90] K.-F. Ssu, W.-T. Wang, and W.-C. Chang, "Detecting Sybil attacks in Wireless Sensor Networks using neighboring information," *Computer Networks*, vol. 53, pp. 3042-3056, 2009.

[91] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: A survey," *Journal of Big Data*, vol. 2, pp. 1-41, 2015.

[92] C. Yang, A. Mason, J. Xi, and P. Zhong, "Configurable Hardware-Effcient Interface Circuit for Multi-Sensor Microsystems," in *Sensors, 2006. 5th IEEE Conference on*, 2006, pp. 41-44.

[93] J. Xi, C. Yang, A. Mason, and P. Zhong, "Adaptive multi-sensor interface system-on-chip," in *Sensors, 2006. 5th IEEE Conference on*, 2006, pp. 50-53.

[94] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, pp. 50-58, 2010.

[95] SamIam. (2015). *The Bayesian network learning tools*. Available: http://reasoning.cs.ucla.edu/samiam/

[96] C. Panos, C. Xenakis, and I. Stavrakakis, "A novel intrusion detection system for MANETS," in *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, 2010, pp. 1-10.

[97] J. Tang, X. Huang, J. Qian, and C. Viho, "A FSM-based Test Sequence Generation Method for RPL Conformance Testing," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom),*

IEEE International Conference on and IEEE Cyber, Physical and Social Computing, 2013, pp. 591-597.

[98] N. Stakhanova, S. Basu, Z. Wensheng, X. Wang, and J. S. Wong, "Specification synthesis for monitoring and analysis of MANET protocols," in *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, 2007, pp. 183-187.

[99] T. Tsao. (2011, Internet-Draft v00: A Security Design for RPL: IPv6 Routing Protocol for Low Power and Lossy Networks http://tools.ietf.org/html/draft-sdt-roll-rpl-security-04.

[100] T. Matsunaga, K. Toyoda, and I. Sasase, "Low false alarm rate RPL network monitoring system by considering timing inconstancy between the rank measurements," in *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*, 2014, pp. 427-431.

[101] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, *et al.*, "RFC 6550: RPL IPv6 Routing Protocol for Low-Power and Lossy Network," *IETF*, 2012.