Key pre distribution in the context of IoT: the RPL new objective function SISLO

Citation: El Hajjar, Ayman (2022) Key pre distribution in the context of IoT: the RPL new objective function SISLO. [Thesis] (Unpublished)

Key Pre Distribution in the Context of IoT: The RPL new Objective Function SISLOF

Ayman El Hajjar

Supervisors: Prof. George Roussos
Prof. Maura Paterson

Department of Computer Science
Birkbeck College, University of London

This dissertation is submitted for the degree of
Doctor of Philosophy

Birkbeck University of London June 2022

Dedication

I dedicate this work firstly to Reema, my wife who has supported me and reminded me every day of the task that I needed to complete and made me believe that I can finish this PhD no matter what.

I also dedicate this work to Danny, my son who kept on giving me hugs and love during this year. He was only one years old when I started this journey and slept many times on my lap while I was reading papers. Without them, I would not have found the courage and dedication to finish this journey.

Finally, I would like to dedicate my work as well to my parents Ziad and Amina who have supported me since the beginning and believed in me, invested in my education in the UK and were there for me whenever I needed them and for my brothers Tarek, Houssam and Bahaa. Without them I would not be the person I am today.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or processional qualification except as specified.

Parts of this work have been published in [1], a summary of experiment results of chapter 3, [2] a summary of experiment results of chapter 4 and [3], a summary of experiment results of chapter 5.

Acknowledgement

I would like to express my gratitude and patience to the supervisory team who have guided me throughout this long journey. It would not be possible to be where I am at this moment both academically and in my career without the support of my director of studies Professor George Roussos and my second supervisor from the Department of Economics, Mathematics and Statistics Professor Maura Paterson.

I would also like to thank the scholarship committee in the department of Computer Science and Information Systems for offering me this opportunity that allowed me to undertake this study.

Finally, I would also like to extend my regards to other researchers and PhD students at the university whom I have shared an office or met regularly and we had many deep or funny discussions and I have formed a good relation with them that hopefully I will maintain for many years to come.

Abstract

The purpose of this thesis is to develop a novel objective function that ensures secure links between all nodes in an Internet of Things network when using the Routing Protocol for Low-Power and Lossy Networks (RPL) and only allow nodes in the network that share a key to join the network.

We propose the Shared Identifier Secure Link Objective Function (SISLOF) to allow only nodes that share a key to join the network and therefore ensuring that all links between the nodes in the network are secure. SISLOF will look at a route that includes all nodes in the network and if a node shares a key with more than one node, it will then choose the node that has a shorter pathway to the root.

We evaluate the overhead of the security keys on the Internet of Things nodes and the routing metrics by measuring the overhead when using first ETX and OF0 objective functions when using either the probabilistic scheme or the deterministic scheme. We then identified that the use of ETX or OF0 with both schemes is not appropriate because of the large overhead it adds on the devices and the link. We show that both ETX and OF0 add a large overhead and they are not suitable to be used with the security schemes. The secure objective function was needed as the existing objective functions add a large overhead on the Internet of Things devices when using two different key distribution schemes to distribute and provide keys between nodes and to create a link. We develop an objective function that only adds nodes that share a key to the routing table without the overhead cost the other objective functions added. We also identify that the probabilistic key distribution scheme outperforms the deterministic key distribution scheme for all objective functions.

The significance of this study is that it has identified the need for an objective function that incorporates the security key distributions for the Routing Protocol for Low-Power and Lossy Networks (RPL) in the Internet of Things networks and the Shared Identifier Secure Link Objective Function (SISLOF) was developed to solve this problem.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Research overview

In this Chapter we outline the research questions for this thesis and we also look at the aims and objectives of this research. We also identify the main contributions from this research. We present the research methodology used and outline the structure for the rest of the thesis. We end with a statement of originality.

1.1 Motivation

The Internet of Things is the next evolution of the Internet which will substantially affect human life. IoT is important because it is the first of its kind that is propelling an evolution of the Internet and smart environment; Everything will be connected and data of our environment and of our physical presence will be used to takes on decision such as setting the thermostat automatically by sensing the temperature of the surrounding environment. It is clear that secure communication between IoT devices is essential and the threats and risks for having an insecure IoT are a lot bigger than for conventional Internet connected devices.

The motivation behind this research is to find a reliable and efficient mechanism for nodes within the IoT and to establish trust by securing end-to-end communication by having a certain pre-distributed key scheme that will enable such communication by the use of a Key pre-distribution scheme (KPS).

1.1.1 DSN & IoT Differences

Many KPS were proposed for Distributed Sensor Networks (DSN). DSN shares a lot of the IoT characteristics as discussed in [5] and can be used as a starting point for this research.

Although both DSN and IoT are considered infrastructure-less networks and operate on an Ad-Hoc basis, many essential characteristics (by definition) between them are not shared. Those characteristics change the whole environment of IoT in comparison with DSN. Distributed Sensor Networks are not able to use classical IP based protocols simply because it is very difficult to allocate a universal identifier scheme for a large DSN and proprietary protocols are usually used to identify unique devices and explained in [6]. A distributed sensor network can operate by itself sending data to a centralized entity in order to monitor the physical conditions of an environment. An IoT network requires one or more devices to act as a sink and to connect the network to other types of networks such as the Internet in order to send data collected. The devices in an IoT network do not need to be the same and all can communicate to complete a specific task.

For that reason, DSN nodes cannot inter-operate and communication between various nodes only exist for routing purposes and to allow data to reach the centralized location. Since IoT nodes are able to inter-operate with the existing Internet infrastructure, each of them needs its own unique identifiable Internet protocol (IP) address rather than a proprietary protocol.

Addressing and identifying nodes in a DSN network presents us with a complete set of challenges that differs in the scenario of an IoT network. The flow of data in a DSN network is most of the time in one direction towards the sink connected directly to the centralized location. The flow of data in an IoT network is bi-directional as a node can either send data to the Internet or receive instructions from another entity. This difference means that the routing protocols used for a DSN network cannot be used in an IoT network. In most applications of DSN networks, route discovery base routing protocols are used; Ad Hoc On Demand Distance Vector (AODV) in [7], Dynamic Source Routing (DSR) in [8] and Optimized Link State Routing (OLSR) in [9]. Each of those protocols have their own characteristics however they all share two important features. All are proprietary protocols and are not IP based protocols but proprietary classless protocols and they only allow route discovery and route establishment messages to be exchanged between nodes in both directions in comparison with the IoT where data can only travel through one direction at a time.

There are some challenges that need to be taken into consideration when implementing the KPS in the context of the Internet of Thing. The use of a suitable symmetric encryption protocol is also essential. Different encryption protocols require different time to decrypt as each will present different limitations in terms of computation and processing speed.

DSN network nodes were assumed to have proprietary unique identifiers simply because they were never intended to be used as part of a large network such as the Internet. This is not a practical solution for the IoT as data is needed to travel between two directions and sometimes directly to the Internet. For that reason, it requires an IP based routing protocol. Most of the conventional devices on the Internet uses the Transmission Control Protocol/Internet Protocol TCP/IP communication suite to identify how data should travel between devices, in which format and using which route. This suite however was not intended to be used with the IoT and it is not suitable for the IoT as the devices that participate in this type of network are considered lightweight resource constraints devices. Some attempts were made to develop a unique addressing scheme for the IoT until most researchers and IoT device manufacturers agreed that devices should use the same addressing scheme as the Internet to make it easier for devices to communicate with the Internet. Using IP protocols in sensor networks simplify the connectivity model as the hierarchy of the devices in the network can be flattened. This also removes the complexity of having devices to translate between proprietary protocols and standard Internet protocols as explained in [10] .

However, the TCP/IP suite was still considered heavy and IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) was created for IoT specifically. 6LoWPAN defines how to layer, transmit and deal with data using IPv6 over low data rate, low power, and small footprint radio networks as identified by IEEE802.15.4 in [11] radio. Routing is a fundamental piece of the overall IPV6 architecture for the Internet of Things. The networks in these environments can be described as Low Power and Lossy Networks (LLN), meaning they often operate with significant constraints on processing power, memory,nd energy translating into high data loss rates and low data transfer rates and instability. The routing protocol for Low Power and Lossy Networks (RPL) introduced in [4] was developed to translate the potential of Internet of Things into reality. The objective of RPL is to organize a network topology with thousands of nodes that are energy-constrained by constructing one or more Destination Oriented DAGs (DODAGs) explained in details in section 2.4, and therefore it is crucial for the speed that large amount of the nodes join the DODAGs with few costs of energy consumption if possible. RPL solves the unique challenges that IoT brings to the exchange of messages between nodes in a conventional DSN network.

The physical nature of the IoT devices makes it difficult to implement security schemes to secure communication between nodes. In an IoT device, limited resources are available such as the limitation of storage capacity and processing power. A KPS used to secure communication between DSN devices assumes the presence of several routes to a node and if a shared key between two nodes does not exist an alternative secured route can be found. This is not the case in the IoT and therefore a large number of keys is needed to ensure that all links between nodes is secure. This will require a large storage space for a large scale IoT network. This solution will present a problem for IoT devices.

The architecture of the IoT, similarly to the DSN is of Ad-Hoc mode (also known as peer to peer). It means that there is no centralized entity that organizes the distribution of the keys between nodes. It also means that all links between any two nodes needs to contain a shared key. This will naturally result in an increase of the number of keys that each node should have to make sure that all links between two nodes are secured by the use of the shared link. This presents us with another challenge as the implementation of any suggested solution will be limited by the storage capacity of devices used regardless of which KPS scheme is used. The difference in how devices communicate in an IoT in comparison with DSN as explained in this section means that devices that do not share a secure key cannot communicate indirectly if a secure route between them cannot be identified when the routing table is formed using RPL. This will lead to several devices in the network not being included in the routing table and thus will not be allowed to join the network.

Secure communication between end to end IoT devices is essential. IoT devices are meant to exchange data from critical infrastructure such as devices in smart cities, smart houses, SCADA systems and other important infrastructure. Those devices will not only be exchanging important data but also participating in automated decision making and this makes the security of the communication between those devices more important. An attacker listening to the communication between those devices, if the devices are communicating in plain text, can simply intercept the message and understand it. For example, a camera device sending a message to a heating source in a smart home, informing the heater that there is no one at home in order for the heating to automatically go off, will give clues to any attacker who is listening to this communication and thus be able to deduct that the house is empty and a theft can take place.

1.1.2 IoT Threat Model for IoT

In this section we will look at the threats on Internet of Things and identify where the research problem that this research is attempting to solve fits. Authors in [12],[13] and [14] categorized the attacks in different categories as shown in Figure 1.1. As we can see from Figure 1.1, several attacks can be mitigated if nodes in an IoT network communicate in a secure way. The motivation to mitigate those threats all at once is because by ensuring that only nodes that share one or more secret key can communicate we ensure that all nodes that have joined that network are genuine and trusted.

In Section 2.5.6 we present the attack surface which are point of entries or boundaries of the Internet of Things systems that can be exploited using the threats identified in this section and relevant to this research problem such as attack surfaces on key distribution, key storage or the process of routing formation and maintenance.

The threats that this research is attempting to solve and shown in the diagram shown in Figure 1.1 and is collected from the various threats we have identified from variuous research papers on the threats and attacks on the Internet of Things. We in blue and gray attacks that are either directly related to the research problem or indirectly related and hence the solution solves. Those attacks are summarized below.

Generalized category threats on IoT identifies threats that do not only exist in IoT environments or multi-layer threats. Security and user privacy are essential to maintain in any network and protecting the confidentiality and integrity of data from violation will prevent devices from leaking private user data and confidential data. Researchers in [15] and [16] have identified that IoT devices have higher chances of leaking private and confidential data due to the lack of reliable authentication, the lack of data encryption and the lack of network access control measures.

Cryptanalytic attacks explained in [17] [18] exploits the weaknesses in the cryptographic algorithm and can result if successful in the attacker discovering the original message. There are several cryptanalytic attacks that all networks can be vulnerable to depending on the cryptographic algorithm used. Cryptanalytic attacks will result in the violation of confidentiality, integrity and availability of data transmitted in such networks. The type of encryption used to encrypt data will be essential to ensure that the IoT secure DODAG is not vulnerable to cryptanalytic attacks. Ensuring that no malicious node can compromise the network will also prevent this type of attacks as devices will not be able to participate in the network in order to carry such attacks. The solution proposed in this research will have a direct impact on mitigating this attack.

Denial of Service (DoS) attacks on IoT devices explained in [19] result in resources exhaustion due to the physical features of the Internet of Things devices such as low processing power and low battery consumption. Resources exhaustion attacks include jamming of communication channels, extensive unauthorized access and malicious utilization of critical IoT resources and those attacks result in operational functionality of IoT devices or non availability which result in disruption of services. 96% of the devices involved in Distributed Denial of Service DDoS attacks were IoT devices and participated in Botnets as discussed in [19] and [20]. Although this attack is out of context of the research and having encrypted data between nodes do not prevent it directly, however some DoS attacks are carried out by malicious nodes that exhaust the resources of other nodes until they crash.

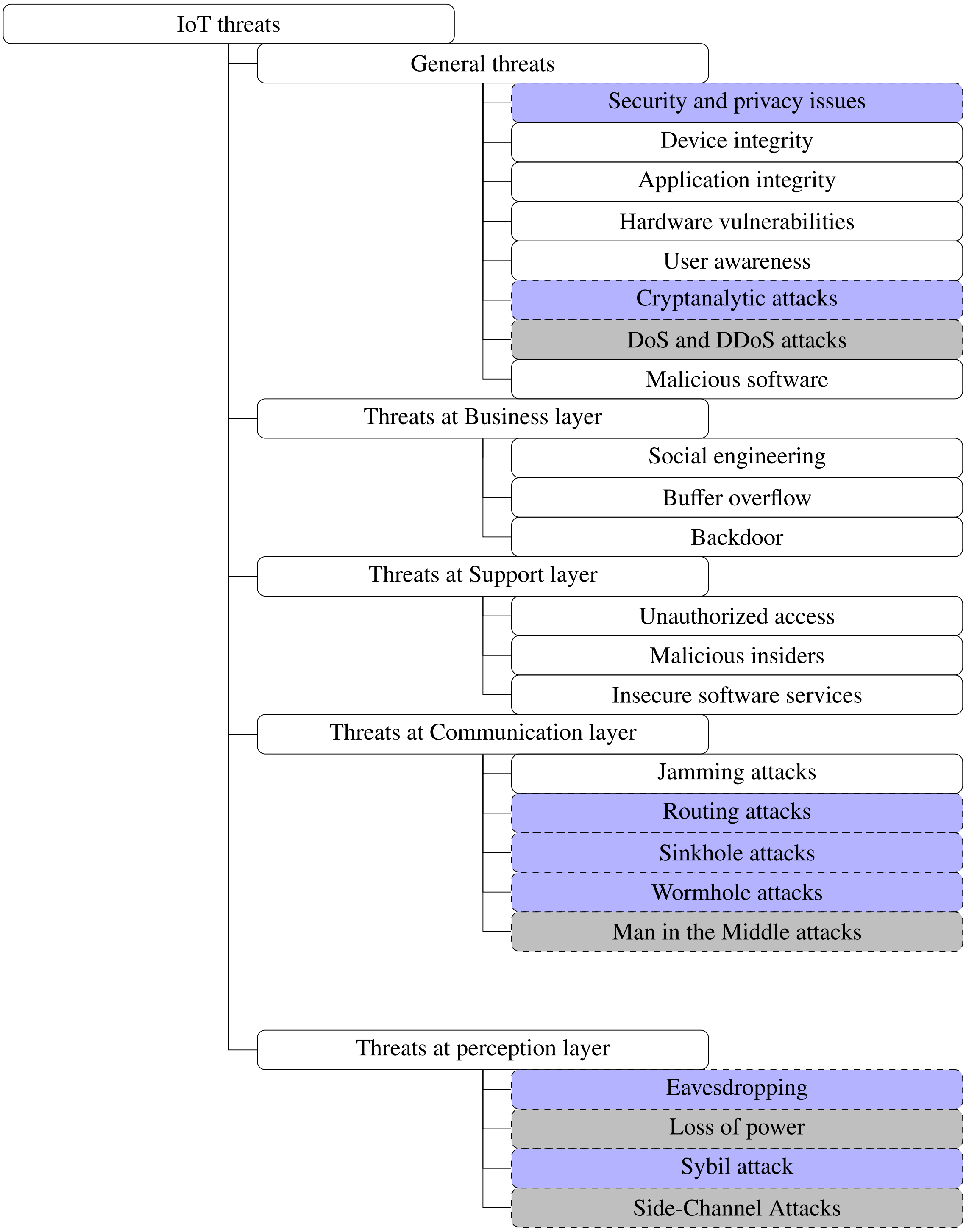


Figure 1.1: IoT threats categorized based on the IoT layers that is affected. For each category, the threats are either considered not related to the research topic (in white background), directly related (Blue background), indirectly related (Gray background)

Securing the routing formation will prevent malicious nodes from joining the network and hence protecting networks against DoS attacks.

Various attacks threaten the Internet of Things routing formation and routing process as investigated in [21], [22], [23] and [24]. IoT RPL DODAG is vulnerable to a selective forwarding attack. In this attack malicious nodes do not participate in transmitting the packets received by it and destroys the routing path of the network by doing so as explained in [25] and [22]. The Blackhole attack explained in [26] is an example of a selective forwarding attack in which a malicious node do not forward any packet and breaks the DAG in the routing table. HELLO flood attacks threaten the RPL DODAG formation process. In this attack when a genuine node utilizes HELLO messages to join a network a malicious node can capture this packet and use it to declare itself a neighbour. In this case, the DODAG Information Object DIO messages can be utilized with strong routing metrics in order to start such an attack as in [25] and leads to the malicious node joining the RPL DODAG. Rank attacks in RPL are other type of attacks in which malicious nodes advertise falsely their rank as discussed in [27] and [28]. Increased rank attack and decreased rank attack are two examples of rank attacks examples in which a malicious node falsely advertise its rank either lower or higher and repeatedly does this in a way that it disrupts the routing topology as nodes will have to regularly update their preferred parent based on the new rank that the malicious node is advertising.

Routing attacks are at the core of the motivation of this research since preventing routing attacks will mitigate several other threats such as preventing malicious nodes from joining the network. Other type of routing attacks discussed in [25] , [29] and [30] are the sinkhole attack and the wormhole attack. In the sink node attack, malicious nodes redirect the traffic of a network to a specific node that acts as a sink node. Several malicious nodes participate in this attack by advertising a particular route that leads to the malicious node that is acting as a sink node. In the wormhole attack investigated in [31] , [32] and [25], the malicious nodes create direct links with each other and force the network traffic data through those links rather than links with intermediate nodes. Sinkhole attack and wormhole attack can be prevented by securing the routing formation process and encrypting the traffic between nodes as it will prevent malicious nodes from joining the network.

Other Man in the middle MiTM attacks discussed in [33], [34] and [15] are defined as a form of eavesdropping in which malicious actors can intercept the traffic exchanged between two nodes and tamper with the exchanged node or use the captured packets to carry on further attacks. Different examples of MiTM can threaten the confidentiality and authenticity of the Internet of Things network such as Neighbor Discovery Protocol NDP poisoning explained in [35] and [36], Address Resolution Protocol (ARP) poisoning identified in [37], replay attacks in [38] and [39] and session hijacking in [40] and [41]. Man in the Middle attacks can be prevented indirectly since encrypted traffic will prevent malicious node from carrying on such attacks and they are unable to decrypt the traffic to get the parameters and values needed to tamper the data in session in hijacking or to replay the traffic.

Threats at perception/physical layer consists of sensors, actuators, computational hardware, identification and addressing of the things. Securing data sensing and data collection in this layer is essential as they are done at this layer as explained in [12]. Threats in this layer are related to the physical aspects of the device such as resources exhaustion that causes battery drainage and loss of power by preventing a node from sleeping or going into saving mode. Malicious actors investigated in [42], and [15] can physically install unauthorized devices in order to sniff the traffic and extract valuable information. Eavesdropping and traffic analysis can go together as the sniffed traffic can be captured and analysed by a network packets analyser to gather information about the nodes and their environment in the network. The solution protect against the threat of eavesdropping since malicious nodes cannot decrypt or understand the context of the captured or sniffed traffic. Loss of power if it is caused by the threat of DoS attacks can be indirectly protected by the proposed solution as it prevents malicious nodes from joining the network in order to generate large amount of traffic and exhausts nodes until the battery is drained. If the loss of power is the result of physical tampering of the devices then this solution will not prevent it.

Sybil Attack investigated in [13], [43] [44] is a form of attack that the IoT networks can be subject to. In this attack a malicious node impersonate one or more genuine nodes in the network and generate fake data and thus violating the trust and confidentiality between the nodes in the network. This attack can be prevented by this solution as the malicious nodes will be prevented from joining the network.

Side channel attacks as defined by [45] is based on side-channel information about the encryption device that are found on the physical device when data is being processed in the perception and physical layers of the device such as information about data processing time or power consumption of the device when encrypting/decrypting various messages and during the computation of different security protocols. This threat can be mitigated indirectly if a strong encryption algorithm is used to prevent malicious actors from data information leaked generated when the encryption and decryption process of the keys takes place.

1.2 Research Question

The research question this thesis is looking to investigate is whether the Probabilistic key pre-distribution scheme (KPS) proposed in [46] and the Deterministic Key pre-distribution proposed in [47] can be used in an Internet of Things (IoT) environment similarly to how they are in used in the context of Distributed Sensor Networks (DSNs).

While looking at the research question, we can deduce several sub questions that need to be answered in order to identify the effectiveness of a key pre-distribution protocol KPS for the IoT. We first need to establish the differences between DSN and IoT in order to assess whether different KPSs schemes used in DSN are suitable for the IoT. This will be done by investigating whether the identified schemes can provide the same security measure without any modification of the parameters used. We will then evaluate the impact of those KPS's schemes use on the IoT devices and networks without any modification. Based on the answer of the previous question, we will be able to identify the required modifications that are needed to achieve the necessary security measures in the context of the IoT with acceptable security performance and an affordable resource usage on its devices. After identifying the required modifications needed, if any, we should look at what can be optimized in the IoT in order to determine the most effective security measure with the least cost in term of resources.

The main objective in this research is to establish a reliable and efficient mechanism for nodes within the IoT to establish trust by a mean of establishing a secure end-to-end communication by having certain pre-distribution key scheme that will enable such a communication. A pre-distribution Key scheme KPS is therefore needed. Not a lot of research was done in this field. Many KPS were proposed for DSN and ZigBee. Both network technologies share a lot of the IoT characteristics and can be used as a starting point for this research. Some of the research was done on securing the communication of between the nodes in the IoT network but not in securing the routing topology formation. To my knowledge, using a Key pre-distribution Scheme in the context of the IoT is something that was not looked at before to secure the routing formation. The research needs to find the answers for the following questions in order to develop/identify the most suitable KPS for the IoT. To achieve our main objective the research needs to find the answer for the following questions:

1. Determine the advantage and disadvantages of using the Probabilistic or Deterministic key pre distribution schemes for distributed sensor networks in the context of the the Internet of Things.

2. Evaluate the performance of the simulated key management schemes for distributed sensor networks on the Internet of Things using the same variables used in the distributed sensor networks to achieve full connectivity and assess if they are enough to achieve full connectivity in the Internet of Things network.

3. Evaluate the overhead of experiments to determine the quality of service obtained from implementing the key management scheme for distributed sensor networks on the Internet of Things.

In order to determine the advantage and disadvantage of using the key management scheme for distributed sensor networks on the Internet of Things a thorough literature review needs to be done on the key management scheme for Distributed Sensor Networks, why it was chosen as a standard, what are the advantages of having a Probabilistic rather than a Deterministic schemesand the performance of a Probabilistic scheme once implemented on a DSN. We will also need to determine what the disadvantages of using this scheme are, in order to understand the limitation of the protocol and the challenges it brings.

An important step before implementing the key management scheme for Distributed Sensor Network on the Internet of Things is to determine the suitable metrics for the internet of things. Metrics such as the key size and the type of encryption used are critical in realizing the quality of service and performance acceptable for the Internet of Things. It is essential to make sure that the size of the key and the encryption used are small enough to fit in the small limited memory of the Internet of things devices and that the encryption and decryption process does not compute a lot of processing power because of limitation in such devices connected to the Internet of things.

After determining the suitable metrics for the Internet of Things, we will implement the key management scheme for distributed sensor networks on the Internet of Things. The implementation will assess how the key management schemes will perform when used in the context of the Internet of Things. This will be done by first simulating those key distribution schemes in various sizes and using different variables such as the number of nodes and number of keys in the pool as defined in Chapter 3 and implemented in Chapters 4, 5 and 6. We will then evaluate how those schemes perform in a real world deployment in Chapter 7. Several Objective functions are used and tested in order to identify which Objective Function is the most appropriate to use with an encrypting routing traffic. An Objective Function discussed in details in Section 2.4 defines how a RPL node selects and optimizes routes within a RPL Instance based on the information objects available. The Objective functions For both simulated environment and real world deployment experiment the following steps will be done:

- Implement the Probabilistic key management scheme in the context of the IoT for distributed wireless sensor network in the simulated environment using RPL with either OF0 or ETX objective functions with the variables identified before.
- Implement the Deterministic key management scheme in the context of the IoT for distributed wireless sensor network in the simulated environment using RPL with either OF0 or ETX objective functions.
- Evaluate the performance of both schemes when simulated and determine if the overhead of the Probabilistic and Deterministic scheme when using either OF0 or ETX objective functions are within the acceptable overhead and do not underpin the performance of the nodes in the Internet of Things network.
- If the overhead found is not acceptable, then create an objective function that can use the key distribution schemes in a more efficient way to reduce their overhead so that nodes in the Internet of Things network can work in an efficient way.
- Evaluate the performance of both schemes in the real world deployment and compare results with results obtained in the simulated environment.

Finally, we analyse the results of simulation to determine the quality of service obtained from implementing the key management scheme for distributed sensor networks on the Internet of Things. We also compare the results of simulation on a test-bed to a real life experiment on a small scale. This will give us a clear idea of how the results from both simulation experiments and test bed experiments differ in terms of quality of service for the IoT.

1.3 Aims & Objectives

The security of the communication links between nodes in the Internet of Things has not been a focus of many research and incorporating securing the communication links of the joining nodes in the DODAG is something that is needed. The aim of this research is to evaluate the performance of secure IoT network using RPL routing protocol with the various objective functions and either a probabilistic or a deterministic scheme.

In order to investigate if KPS is a viable approach, the advantages and disadvantages of using it in the context of the IoT will be assessed.

An important step before implementing the key management scheme for Distributed Sensor Network on the Internet of Things is to determine suitable parameters. Once the suitable parameters for using KPS on an IoT network are determined, a validation will be done by mean of implementing and evaluating the security of the IoT network using those identified parameters on KPS. We will also compare the results of simulation on a test-bed to a real life experiment on a small scale. This will give us a clear idea of how the results from both simulation experiments and test bed experiments differ in terms of quality of service for the IoT.

In this section, objectives are derived from the aims and an explanation of how those objectives will be achieved is presented.

1. Determine the similarities and the differences between the wireless sensor networks, the distributed sensor networks and the Internet of Things to provide a clear classification of each of those networks in term of the number of communication links between nodes and the routing formation process.

2. Investigate the use of the Probabilistic key pre distributed scheme to achieve full connectivity in DSN in the context of the IoT and identify the impact of using this scheme with the variables used on the routing formation and nodes performance.

3. Investigate the overhead performance of both KPSs on the the routing formation and the nodes performance when using ETX and OF0 objective functions.

4. Develop an objective function that uses either Probabilistic or Deterministic schemes in the routing formation in order to only allow nodes that share a key to form a leaf in the DODAG of the network.

5. Examine the overhead performance that the developed objective function using Probabilistic or Deterministic schemes to identify which KPS is more suitable.

1.4 Contributions

At the end of this thesis, the contributions listed below were made and all contribute to the understanding of how Key Pre Distribution can be used in the context of the Internet of Things.

- **The impact of the use of Key Pre-Distribution schemes on different variables in the IoT:** We developed in Chapter 3 a model that outlines the cost of using KPS in the context of the IoT to allow researchers to quantify the cost of KPS security for any size of IoT network using any device. This was achieved by identifying the different variables in an IoT network and the overhead the use of the KPS result on the IoT nodes.

- **Key distribution schemes in the context of IoT using DSN variables**: We have identified in Chapters 4 and 5 that the results obtained when used those schemes in the context of the Internet of Things does not achieve the same results obtained when used in the context of the Distributed Sensor networks. This is due to the main differences in how the communication links are formed between nodes as the nature of the routing protocol RPL that only allows one link to exist between two nodes and the routing table formation.

- **Comparison of the Key distribution schemes performance in the context of the IoT**: We have identified in Chapters 4 and 5 that neither Probabilistic or Deterministic schemes can be used in the context of the IoT while using the routing protocol RPL in its current form without any modification. This was observed when the overhead of both schemes was too high on the IoT nodes.

- **Preferred key distribution schemes performance in the IoT**: We have also identified that the Probabilistic key distribution scheme is more suitable to use in the context of the IoT due to the overhead that limited neighbouring nodes adds to the computation of the preferred parent and the route to the root node in a DODAG in the IoT network. When using the Deterministic key distribution scheme, the FMAP mutual authentication and the voting process in this scheme determines the lack of trust and the mutual agreement between nodes and result in some nodes discarded due to the lack of trust between nodes. This in term results in an addition in a large overhead on the routing formation process and the link quality between nodes when used in the context of the IoT.

- **Shared Identifier Secure Link Objective Function SISLOF**: We demonstrated that our new proposed objective function (SISLOF), the Shared Identifier Secure Link Objective Function SISLOF allows RPL to only create a routing table between modes that can establish a secure link and outperforms other objective functions when using either Probabilistic or Deterministic key distribution schemes. This is the main contribution of this research that identified how a key distribution can be integrated in the routing process of the IoT to only force nodes that share a secure key between each other and form a leaf in the DODAG.

1.5 Research Structure

The remainder of this thesis is divided into several Chapters.

Chapter 2: literature Review- This Chapter introduces the four main fields of the thesis and look at previous research work that is relevant to the topic. An introduction to the IoT is presented with a detailed explanation of the differences between IoT and DSN networks. The architecture of 6LoWPAN communication suite is explained and an explanation of how the 6LoWPAN differs from TCP/IP communication suite used by the devices connected to the conventional Internet. Following the 6LoWPAN architecture, a thorough explanation of the routing protocol RPL and its objective functions is provided. Finally various key pre distribution schemes that provide secure communication between devices are explored and an assessment of how some of those schemes were used in the context of DSN is shown.

Chapter 3: Simulation Experiments- This Chapter describes the simulation platform Cooja used on Contiki Operating System. It also explains the mathematics behind choosing the variables, identifying the number of nodes for each simulation and how simulations will be validated in comparison with the mathematical formula.

Chapter 4: Probabilistic key pre-distribution- It looks at different Probabilistic key pre-distribution schemes and implements the key pre-distribution protocol proposed in [46] in the context of the IoT. The Chapter outlines the various assumptions made such as the communication security constraints and key management constraints. It then studies the key distribution and revocation methods proposed and how it achieved full connectivity by only having 50% of the devices sharing keys. The Chapter then continues by experimenting with the different number of keys in the key ring in order to achieve full connectivity in the context of the IoT. the impact of securing the IoT network is then evaluated based on the number of nodes that are unable to join the network because of their inability to either find a secured route or sharing a key with the direct branch of the RPL routing tree.

Chapter 5: Deterministic Key pre-distribution- This Chapter looks at the different Deterministic key pre-distribution schemes and implements the key pre-distribution protocol proposed in [47] in the context of the IoT. It provides an explanation of the network environment and assumptions made by [47] for DSN networks and how they were taken when used in the context of the IoT. An explanation of the algorithm and how it works in terms of the various phases to identify secure routes between first 2 hop paths and beyond. The Probabilistic key pre-distribution algorithm is then evaluated and analysed in terms of its performance, the network topology and the number of keys needed when used in the context of the IoT.

Chapter 6: Shared Identifier Secure Link Objective Function (SISLOF)- In this Chapter, a modification of the RPL routing protocol is proposed to ensure that only nodes that share a suitable key can join the RPL routing table. This will ensure that all IoT network nodes connect in a secure method. SISLOF uses the concept of key pre-distribution proposed in [46] in the context of the Internet of Things. The metrics used in the context of the IoT are identified from previous Chapters and evaluated in the context of the IoT when SISLOF Objective Function is used.

Chapter 7: Hardware experiment- In this Chapter, simulation of the experiments carried out in both Chapter 5 and Chapter 6 is done using real nodes in a smaller environment. The variables are identified based on calculations provided in the previous experiments but in relation to a smaller number of nodes. The results are then compared and evaluated in order to validate the results in a real-life environment in comparison with a simulated environment.

Chapter 8: Analysis, conclusion and future work- this provides an analysis of the Probabilistic approach and the Deterministic approach for key pre-distribution in the context of the IoT when RPL is used with either the RPL OF0 or RPL ETX in comparison with RPL with the proposed SISLOF. It also summarizes the finding of this research and assesses whether the results obtained in comparison with the aims and objectives identified at the beginning of this research have been achieved.

1.6 Assumptions and Limitations

Several assumptions are be made for this research as they are either out of context of the research problem or they were needed to ensure experiments are as close as a real life scenario as possible. We will list in this section the main assumptions for the whole research.

Assumption 1- Key pre-distribution: We first assume that keys were distributed using a method outside the context of this research. This could be when nodes were manufactured or using a centralized entity that generate keys and distribute them randomly to all nodes. We also assume that all nodes in the network are friendly and none of them are malicious in the meaning that all devices joining the network are authorized to do so. We discuss in details this assumption in Chapter 3 and provide a mechanism to revoke keys and initiate routing formation again if one node is compromised.

Assumption 2- Nodes distribution: Nodes are distributed in a random method however we restricted the environment setup to 250*250 meters to ensure that nodes can still communicate. We also assumed that nodes have a range of 50 meters each in small networks and 25 meters in large networks. If a node was out of reach for the whole experiment, we then generate new nodes locations in the simulation environment until each node signal can reach one other node at least.

Assumption 3- Network and pool size: We assumed in all experiments that the maximum pool size needed is of the same value of the network size however, we experiment with various pools starting from small pool sizes until we reach the network sizes for a pool. For example, running a network larger than 100 nodes with a pool size larger than 100 keys proved before that it will yield to very low shared keys percentage and to achieve connectivity very large rings size will need to be used which is an unrealistic approach and therefore we stopped at 100 keys in the pool for such networks.

Assumption 4- Parameters and performance: When comparing the different objective functions, we consider one outperforms the other if the value is larger for the ring size, number of securely connected nodes in a DODAG and the number of neighbours that a node shares a key with. We consider one underperforms the other if the parameter value compared is larger for the total number of RPL control messages generated by all nodes, the total power consumption for all nodes, the time the DODAG needs to converge and the average time a packet needs to reach the root node.

Assumption 5- Identity uniqueness: Following the assumptions made in [47] for the identity uniqueness and since the node fingerprint is out of context for this research, we assume that each node can assume a unique identity in the network that all other nodes agree on. This is to prevent a malicious node from existing in the network before the routing formation process even starts.

Assumption 6- Fingerprinted Mutual Authentication Protocol (FMAP): Following the assumptions made in [47] for the fingerprinted mutual authentication between two nodes, we assume that each node has the ability to distinguish when computing the FMAP a genuine identity fingerprint from a fake one.

Assumption 7- Time to converge duration: We assume that for a DODAG to become stable and no changes occur a 24 hours duration is needed. This is not a realistic period as the DODAG should become stable in lot less time but we wanted to identify if the overhead of the rings and the encryption/decryption process makes a node changes its preferred parent duration after certain time.

Assumption 8- Keys and identifiers sizes: We assume that 64 bits keys and 32 bits identifiers are realistic sizes considering the number of nodes we will be using in our experiments and the sizes of the pools.

Assumption 9- Keys and identifiers sizes: We understand that the keys and identifiers are small and do not provide a high level of security if data are being transmitted, however, routing information are being sent between each leaf in the routing table using different set of keys and identifiers.

Assumption 10- Keys and identifiers sizes and hardware limitations: IoT hardware used in this research has a limitation in term of the storage space as explained in Section 3.3. Other IoT devices have more space and therefore different keys and identifiers sizes can be used.

Assumption 11- Keys generation: The 64 bits keys generated in each pool are outputs of the 40 bits key and the 24 bits Initialization vector.

Assumption 12- Symmetric encryption algorithm: This research does not go further in assessing the impact of the encryption algorithm used as the main interest of the research is to identify the impact of using symmetric encryption in the process of the DODAG formation impacts the topology and the nodes performance. Therefore RC4 is used to encrypt and decrypt the keys even though RC4 is considered insecure however it is known for it is simplicity and speed. For this reason, one IV and the 40 bits are used together in one key and we do not generate a new IV for each packet. This is only to assess the impact of securing the DAG formation. Even if the IV changes, it will have no impact on our experiments since the size of the IV is fixed.

1.7 Ethical Consideration

There are no ethical consideration as this research does not involve any tests on humans or any exchange of private and confidential information. All experiments are stopped before the data exchanges occur since the research question only focuses on the route formation.

Chapter 2

Literature Review

In this Chapter we will introduce the various topics related to this research. An emphasis will be made on the main topics such as the Internet of Things, 6LoWPAN IoT protocol, RPL routing protocol and the different security approaches to secure Internet of Things and different approaches to use key distribution to distribute keys on the Internet of Things.

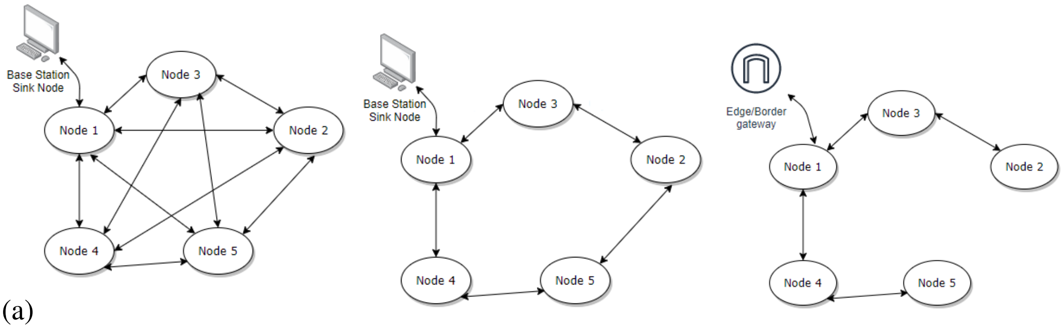
2.1 WSN and DSN

Wireless Sensor Networks (WSN) development like many of the advanced technologies started with the military in the 1950s [48]. With the ever increasing capabilities of low power sensor nodes which include sensing, data processing and communicating, Wireless Sensor Networks WSN was realised based on the collaborative effort of a large number of sensor nodes [49] and adopted in many applications. An example of such early applications was a network of sensors called Sound Surveillance System (SOSUS) [50] that was developed by the United States military to detect and track Soviet submarines. Distributed Sensor networks (DSN) on the other hand is a variation of the WSN that was created in the 1980s to explore the challenges in implementing distributed/wireless sensor networks.

WSN and DSN share many properties and characteristics with IoT networks such as the intrinsic properties of the sensor nodes that those networks are composed of. In all networks, nodes are lightweight, energy efficient and low power devices.

The differences between WSN, DSN and IoT can be summarized by two main differences, first how the nodes connect with each other and report to the sink node or gateway and the number of connections between the different nodes in the network. An example of how five nodes form a network in the different networks is shown below in Figure 2.1. Wireless and

Distributed Sensor networks can take the form of different physical topologies outlined in [51] and can be summarized by three topologies decentralized self organizing, centralized architecture and grid networking techniques.



(a)
Decentralized WSN representation

(b)
DSN representation

(c)
IoT representation

Figure 2.1: Five nodes physical topology comparison for WSN, DSN and IoT networks. Each node in Wireless and Distributed Sensor networks can have one or more links. Distributed sensor networks establish enough links to have a route to the sink node. IoT nodes establish nodes with preferred parent to reach root node.

In order to transform WSN into a viable technology to make the IoT vision cost-effective and deployable, authors in [52] claim the need for middleware-layer solutions fully compliant with accepted standards (or largely adopted specifications). This in fact is essential to allow sensor nodes in IoT to communicate with the Internet to process its data.

2.2 Internet of Things

Internet of Things (IoT) will substantially affect human life and is important because it is the first of its kind that is propelling an evolution of the Internet and smart environment– an evolution that will lead to innovative applications that have the ability to revolutionize our lives and our surroundings.

The vision of having a variety of physical elements “Objects” and “things” connected to the Internet is what forms the IoT. In the conventional Internet, most of the devices connected to the Internet were used directly by humans and needed a direct interaction from a human being to be able to generate data. The IoT vision enabled objects and things to interact with an external entity and send data without the interference of a human. No human participation is needed and objects are able to take decisions based on data received, sent or generated.

Thus the term of the Internet of Things explained in [53] is now considered as a global network which allows the communication between human-to-human, human-to-things and things-to-things that is anything in the world by providing a unique digital identity to each and every object .

The idea is that all objects connected to the IoT will contain embedded technology, allowing them to interact with internal states or an external environment. Those objects will be able to sense and communicate thus changing how and where decisions are made and who makes them. [54]

The IoT is an emerging technology closely related to other research areas like Peer to Peer Networking, Mobile computing, Pervasive or Ubiquitous computing, Wireless Sensor Networks, Cyber Physical Systems, Real Time Analytics, etc. Technologies like ZigBee and Wi-Fi Direct can be widely deployed to achieve the notion of smart cities, eventually achieving a globally integrated smart world. However, there are ongoing issues like architecture design, hardware design, cost accountability, identity, privacy, and security issues for building new devices and solutions in IoT [55].

The applications and usage of the Internet are multifaceted and expanding on a daily basis. The Internet of Things (loT), Internet of Everything (loE) and Internet of Nano Things are new approaches for incorporating the Internet into the generality of personal, professional and societal life [56].

Applications of IoT encompasses medical implants, alarm clocks, wearable systems, automotives, washing machines, traffic lights, and the energy grid. It is expected that 50 billion devices will be interconnected by 2030. Having this huge Global Network will result in the generation of a huge unprecedented amount of data.

Internet protocols have always been considered too heavy for sensor networks and thus the 6LoWPAN protocol stacks were created [57]. 6LoWPAN concept originated from the idea that “the Internet Protocol could and should be applied even to the smallest devices” and that low-power devices with limited processing capabilities should be able to participate in the Internet of Things [11].

2.3 6LoWPAN

To achieve the vision of the Internet of Things, a review of the currently used Internet protocols and standards was needed. The Internet Protocol (IP) was always considered a protocol for Local Area Networks, Wide Area Networks, PCs and servers. The IP protocol was not intended to be used with Wireless sensor networks, Personal Area Networks and the sensor itself. The main reason why it was not intended to be used is that the IP is too heavy for those applications. Sensor networks are meant to be lightweight resource-constraints devices.

However, recently there has been a rethinking of the many misconceptions about the IP. The main discussion was to answer this question “why invent a new protocol when we already have IP” thus the development and standardization of 6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks) was carried out. A simple 6LoWPAN architecture is shown below in Figure 2.2 and outlines the basic concept of connecting low power devices in a 6LoWPAN network with a conventional IPv4/v6 network by using an edge router.

6LoWPAN technology realizes the IPv6 packet transmission in the IEEE 802.15.4 based WSN. And 6LoWPAN is regarded as one of the ideal technologies to realize the interconnection between WSN and Internet which is the key to build the IoT [58].

6LoWPAN defines how to layer, transmit and deal with data using IPv6 over low data rate, low power, and small footprint radio networks 6LoWPAN as identified by IEEE802.15.4 radio. 6LoWPAN protocols resides between the data link layer and the network layer. The adaptation of the full IP format and the 6LoWPAN is performed by the edge router that translates conventional IP traffic to 6LoWPAN traffic as is shown in Figure 2.3 in relation to an IPv6 stack.

Using IP protocols in WSNs simplifies the connectivity model, as the hierarchy of the devices in the network can be flattened. This also removes the complexity of having devices to translate between proprietary protocols and standard Internet protocols. [10]

IoT applications are implemented using a wide range of proprietary technologies which are difficult to integrate with larger networks and Internet-based services. Where as the 6LoWPAN approach is an IP based one, these devices can be connected easily to other IP networks which doesn’t require any translation gateways or proxies, and which can use the existing network infrastructures [59].

It is normal to assume that using IP is too heavy in terms of code size, protocol complexity, required configuration infrastructure or head and protocol overhead. Implementation of 6LoWPAN can easily fit into 32Kb flash memory parts which is suitable for the Internet of Things devices and wireless Networks. 6LoWPAN uses the IPv6 thus the need for configuration servers such as DHCP and NAT is not present as the IPv6 has the Zero Configure and Neighbour Discovery capabilities. The use of IPv6 also allowed the protocol to define a unique stateless header compression mechanism for the transmission of IPv6 packets in as few as 4 bytes.

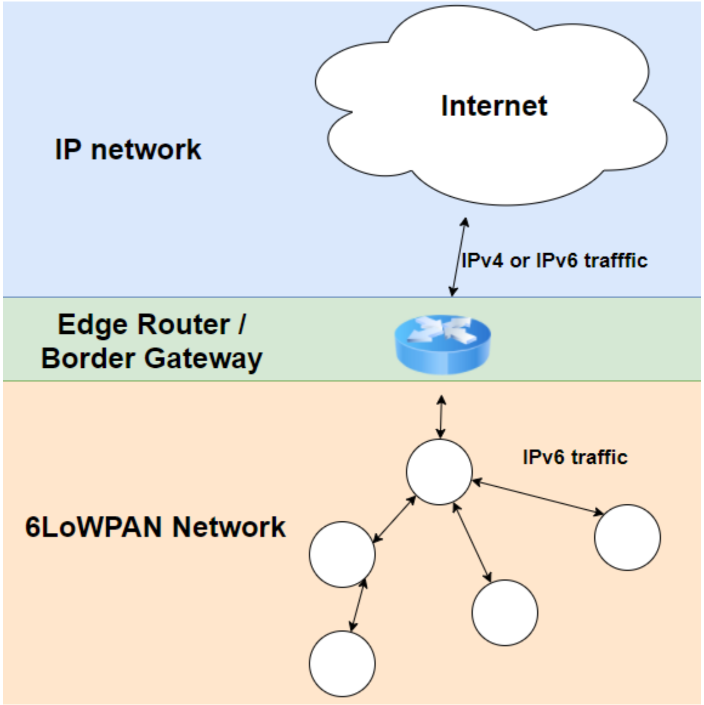
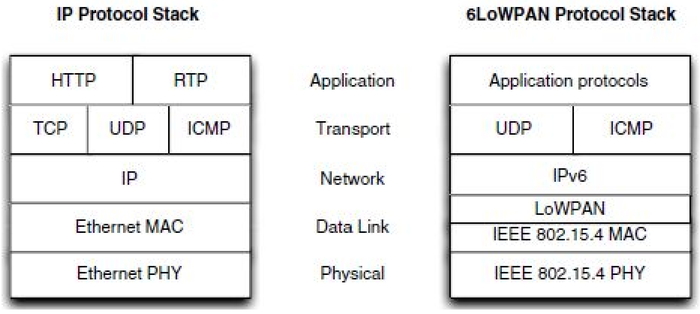


Figure 2.2: The 6LoWPAN simple architecture comprises the IoT network layer, the edge router and the connection to the Internet where the data collected from lower layers are analysed and processed .



IP Protocol Stack				6LoWPAN Protocol Stack	
HTTP		RTP	Application	Application protocols	
TCP	UDP	ICMP	Transport	UDP	ICMP
IP			Network	IPv6	
Ethernet MAC			Data Link	LoWPAN IEEE 802.15.4 MAC	
Ethernet PHY			Physical	IEEE 802.15.4 PHY	

Figure 2.3: IP and 6LoWPAN protocol stacks as presented in 6LwPAN the wireless Embedded Internet by Zach Shelby and Carsten Bormann in [57]. The representation of each layer in the 6LoWPAN shows how the logical communication between the layers at the same level can be interpreted. i.e. Communication between the IP network layer and IPv6.

A key attribute to 6LoWPAN is the IPv6 (Internet Protocol version 6) stack, which has been a very important introduction in recent years to enable the IoT. IPv6 provides a basic transport mechanism to produce complex control systems and to communicate with devices in a cost-effective manner via a low-power wireless network.

The challenges to develop Internet of Things applications using 6LoWPAN stack similarly but with more complexity and can be identified specifically to routing and security of all nodes on the network.

2.4 Routing

Routing is a fundamental piece of the overall IPv6 architecture for the Internet of Things. It became clear as intelligent devices were proliferating into all aspects of life, that a new routing protocol would be required for devices on the smart grid as well as other smart devices operating in harsh environments such as smart grids, manufacturing plants, commercial buildings, and on transportation networks. The networks in these environments can be described as Low Power and Lossy Networks LLN, meaning they often operate with significant constraints on processing power, memory and energy translating into high data loss rates, low data transfer rates and instability. Routing Protocol for Low-Power and Lossy Networks (RPL) is a routing protocol on IPv6 that will translate the potential of Internet of Things into reality.

As of 2011, RPL has been deemed ready by the IETF as a proposed standard RFC. The objective of RPL is to target networks which comprise of thousands of nodes where the majority of the nodes have very constrained resources. RPL protocol consists of routing techniques that organize networks in units called Directed Acyclic Graphs (DAG). DAG is structure where all nodes are connected but there is no available round trip path from one node to another[60].

The DAG structures used in RPL are called Destination Oriented Directed Acyclic Graph (DODAG). The DODAG starts at the root node or sink. The root node is initially the only node that is a part of the DODAG, until it spreads gradually to cover the whole IoT network as DODAG Information Object DIOs are received down in the network. In a converged IoT network, each RPL router has identified a stable set of parents, each of which is a potential next hop on a path towards the root of the DODAG as well as the calculated rank for each preferred parent for each node.

When a router needs to decide on the preferred route to use and on the preferred parent, it will emit DODAG Information Object (DIO) messages using link local multicast thus indicating its respective rank in the DODAG (usually the distance to the root is considered the metric “hop count”). All routers will do the same and each router will receive several DIO messages. Once it receives all DIO messages, it will calculate its own rank and select its preferred parent and then itself start emitting DIO messages.

Since RPL is a Distance Vector routing protocol, it restricts the ability for a router to change rank. A router can freely assume a lower rank but it can assume a higher rank, it is restricted to avoid count to infinity problem. For a router to assume a greater rank, it has to ask the root to trigger global recalculation of the DODAG by increasing a sequence number DODAG version in DIO messages. The protocol tries to avoid routing loops by computing a node’s position relative to other nodes with respect to the DODAG root. RPL is mostly communication between multipoint to point routes from the sensors inside the LLN and towards the root. RPL by way of the DIO generation provides this as upward routers.

Downward routes are only used by parents to issue Destination Advertisement Object (DAO) messages, propagating as unicast via parents towards the DODAG root. In RPL routers two modes exist one that is non storing mode, where an RPL router originates DAO messages, advertising one or more of its parents and unicast it to the DODAG root. The root once it receives all DAOs from all routers, it can use source routing for reaching advertised destinations inside the LLN. The second mode, the storing mode, where each RPL router on the path and the root records a route to the prefixes advertised in the DAO and the next hop.

A routing metric is a quantitative value used to find the cost of a path and helps in making the routing decision in case there are different routes available.e In Low power Lossy Networks a metric is a scalar used to find the best path according to the objective function.

2.4.1 RPL Messages

To understand the messages of RPL and how they propagate over a RPL DODAG, we need to first look at how the messages of RPL are sent. RPL messages typically exist in an IEEE802.15.4 network. The data frame of the IEEE 802.15.4 encapsulates a compressed header of the IPv6 as shown in Table 2.1 and the payload shown in figure 2.4. The compressed header of IPv6 is used since a full IPv6 packet does not fit in an IEEE 802.15.4 frame [61]. The IEEE802.15.4 standard specifies a maximum transmission size (MTU) of 127 bytes, yielding about 122 bytes of actual Media Access Control (MAC) payload [62]. The payload also contains the ICMPv6 control message contained with the IP datagram, also shown in figure 2.4. The type of messages in ICMPv6 is set to 155 when RPL control messages are being sent [63]. Thus an IPv6 header compression is used, encapsulated in the IEEE802.15.4 header as per IEEE802.15.4 specifications in [64]. The IPv6 compressed header of IEEE802.15.4 header is of 5 bytes in size and shown in Table 2.1.

Table 2.1: Size of the different fields of the IEEE802.15.4 frames This is encapsulated in the IPv6 compressed header.

Name of Field	Size in bytes
LOWPAN_IPHC Base Encoding	2 bytes
Context Identifier Extension	1 byte
Next Header	1 byte
Group ID to identify all-RPL-nodes multicast address	1 byte

RPL messages are considered part of the data frame message and they are sent in the payload of an 802.15.4 packet. Control of RPL and the order for a root to form a DODAG and for a node to join a DODAG are shown below :

1. DODAG Information Solicitation message (DIS) (2.4.1)
2. DODAG Information Object (DIO) (2.4.1)
3. Destination Advertisement Object (DAO) (2.4.1)
4. Destination Advertisement Object Acknowledgement (DAO-ACK)(2.4.1) - Optional

DODAG Information Solicitation (DIS)

The DODAG Information Solicitation (DIS) message shown in figure 2.5 as per the definition of RPL messages in [4] may be used to solicit a DODAG Information Object from a RPL node. Its use is analogous to that of a Router Solicitation as specified in IPv6 Neighbour Discovery. A node may use DIS to probe its neighbourhood for nearby DODAGs.

DODAG Information Object (DIO)

A DIO base object structure shown below in Figure 2.6, as per the definition of RPL messages in [4] consists of 24 bytes. This is followed by the route information bytes and metric container bytes.

The RPLInstanceID is an 8 bits field set by the DODAG root that indicates which RPL instance the DODAG is part of. The version number is set by the DODAG root and the rank is a 16 bit unsigned integer indicating the DODAG Rank of the node sending the DIO

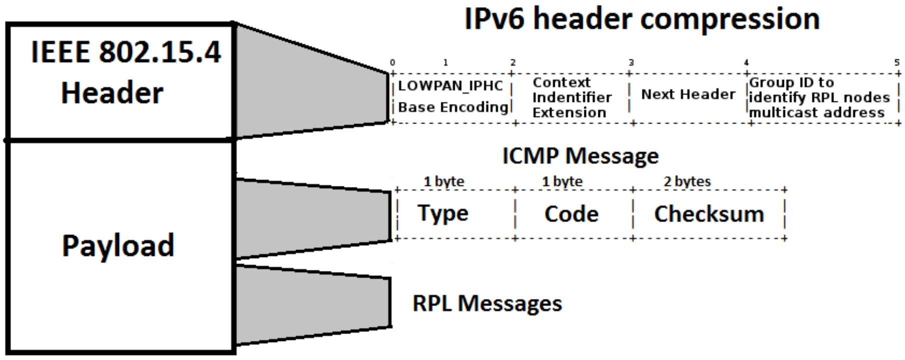


Figure 2.4: IEEE802.15.4 frame with the header and the payload sizes as defined by the 802.15.4 specifications.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |   Reserved    |   Option(s)...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.5: DIS base object frame with the 8 bits unused field reserved for flags. This field is ignored by the receiver and set to zero by the sender. the reserved and the option fields are ignored by the receiver.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RPLInstanceID |Version Number |             Rank              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|G|0| MOP | Prf |     DTSN      |     Flags     |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                            DODAGID                            +
|                                                               |
+                                                               +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.6: DIO message embedded in a 6LoWPAN frame.

message. This defines how the nod receiving the DAO will decide how it will respond to the DIS message. The DODAGID is a 128 bit IPv6 address set by the DODAG root that uniquely identifies a DODAG. The DODAGID must be a rootable IPv6 address belonging to the DODAG root as defined in [4].

The DIO message shown in Fig. 2.6 is embedded in the payload of the IEEE 802.15.4 data frame and takes 80 bytes as defined by Routing Over Low power and Lossy networks (ROLL) in ROLL and shown in Table 2.2 below.

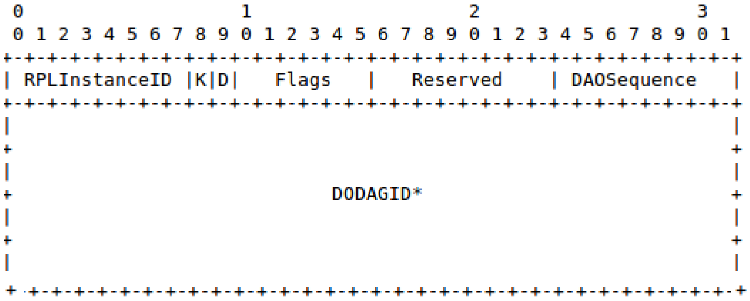
Table 2.2: DIO message fields

Name of Field	Size in bytes
DIO Base Object 2.6	24 bytes
DODAG Configuration Option	16 bytes
Route Information Option	24 bytes
Metric Container	16 bytes

The metric container shown in Table 2.2 takes 16 bytes from the IEEE802.15.4 message. This consists of 2 bytes for "type and option length", 6 bytes for “ETX metric object” and 6 bytes “ETX constraint object”

Destination Advertisement Object (DAO)

A DAO base object format shown below in Figure 2.7 as per the definition of RPL messages in [4] consists of 24 bytes. This is followed by the route information bytes, metric containers bytes and other IPv6 bytes.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RPLInstanceID |K|D|   Flags   |   Reserved    | DAOSequence   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                            DODAGID*                           +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.7: Destination Advertisement Object (DAO) Base Object

The structure of a DAO message shown below in Table 2.3 is 60 bytes.

Table 2.4: DAO-ACK message fields

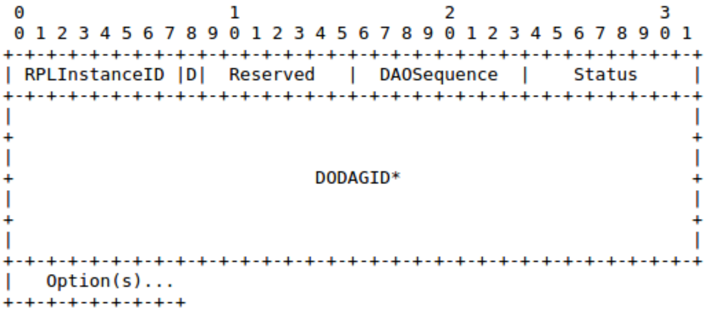
Name of Field	Size in bytes
DAO-ACK Base Object	20 bytes
DODAG Configuration Option	16 bytes
Route Information Option	24 bytes

Table 2.3: DAO message fields

Name of Field	Size in bytes
DAO Base Object (Figure 2.7)	20 bytes
DODAG Configuration Option	16 bytes
Route Information Option	24 bytes

Destination Advertisement Object Acknowledgement (DAO-ACK)

The DAO-ACK message shown in Figure 2.8 as per the definition of RPL messages in [4] is sent as a unicast packet by a DAO recipient (a DAO parent or DODAG root) in response to a unicast DAO message. It consists of 20 bytes. This is followed by route information bytes, metric containers bytes and other IPv6 bytes.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RPLInstanceID |D|  Reserved    |  DAOSequence  |    Status     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                            DODAGID*                           +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Option(s)...
+-+-+-+-+-+-+-+-+
```

Figure 2.8: Destination Advertisement Object Acknowledgement (DAO) Base Object

The 69 bytes of the DAO-ACK message are shown in Table 2.4

2.4.2 RPL Routing Metrics & Constraints

For a DODAG to be constructed, the root will need to first broadcast a DODAG Information Object (DIO) message, discussed in details in Section 2.4.1 to all its neighbours. This DIO message will propagate through the network. Each node that receives a DIO message will consider the sender node a preferred parent to reach the root node until it receives another DIO message with better metrics to reach the root from another node [4]. The DIO message contains the DAG Metric Container option that is used to report metrics along the DODAG. The DAG metric Container may contain one specific metric or various numbers of metrics and constraints as chosen by the implementer [4]. Should multiple metrics and/or constraints be present in the DAG Metric Container, their use to determine the "best" path can be defined by an Objective Function (OF).

Directed Acyclic Graph (DAG) that attempts to minimise path costs to the DAG root according to a set of metrics and Objective Functions. This is one of the known requirements of RPL, and other data-path usage might be defined in the future. The graph is constructed by the use of an Objective Function (OF) which defines how the routing metric is computed. In other words, the OF specifies how routing constraints and other functions are taken into account during topology construction. There are circumstances where loops may occur and RPL is designed to use a data-path loop detection method.

The Routing Metrics and Constraints for RPL are defined in [4]. Those metrics and constraints are used in addition to other variables together and identified as OCP 0 for Objective Function Zero (OF0). When the DAG Metric container contains a single metric, called an aggregated metric, that adjusts its value as the DIO message travels along the DAG. A node decides on its preferred parent and thus its rank based on this single rank only [65]. For example if the node Energy metric is aggregated along paths with an explicit Min function. The best path is selected through an implied Max function because the metric is Energy and thus the node with the highest Energy is selected as preferred parent. However, when a DAG Metric Container contains several metrics, then they need to be used in the order of criteria to be achieved. Each Metric criterion will be first met before moving to the next metric when deciding on a rank of a node (preferred parent). Several Metrics/Constraint Objects exist. In this section, the Metrics and Constraint Objects are discussed.

Each of the objects below is a metric that can be considered a criterion in selecting a preferred parent. When chosen, it will be defined in the DAG Metric Container. Only one object of each metric can exist in the DAG Metric Container. Those metrics objects fall into two categories:

1. Node Metric/Constraint Objects in Section 2.4.2
2. Link Metric/Constraint Objects in Section 2.4.2

Node Metric/Constraint Objects

Node Metric/Constraint Objects are metrics or constraints related to nodes such as node processing power, node memory, congestion situation, node energy (e.g. In power mode, estimated remaining lifetime and hop count to reach the node). Several metrics exist to calculate those criterias

1. Node State and Attribute Object (NSA): The NSA object is used to provide information on node characteristics. Those characteristics of node state and attribute are defined by an 8 bit flag. This flag can have the value 'A' flag or '0' flag. 'A' flag means that applications in this node may use aggregation node attribute in their routing decision to minimize the amount of traffic on the network. '0' flag means that node workload may be hard to determine and express in some scalar form. Node workload will then be set based upon CPU overload, lack of memory or any other node-related conditions.

2. Node Energy Object: The Node Energy Object is used as a metric when it is desirable to avoid selecting a node with low energy. Power and energy are clearly critical resources in most LLNs. Node Energy Object is calculated by determining the node Energy Consumption needed for each node [66].

   $$EE = \frac{Power_{now}}{Power_{max}} \times 100$$

   Where EE is the energy estimation for each node

3. Hop Count Object (HP): The Hop Count Object (HP) is used to to report the number of traversed nodes along the path. The HP object may be used as a constraint or a metric. When used as a constraint, the DAG root indicates the maximum number of hops that a path may traverse. When that number is reached, no other node can join that path. When used as a metric, each visited node simply increments the Hop Count field.

Link Metric/Constraint Objects

Link Metric/Constraint Objects are metrics related to links connecting nodes together such as link quality, link latency, throughput and reliability. Similarly to the Node Metric Objects, only one of each of the objects discussed below can be used at a time in the DAG Metric Container. Several link objects exist to calculate those criteria.

1. Throughput: The throughput is the amount of data moved successfully from one point in the network to another in a given time period. The throughput object is calculated

by calculating the estimated actual throughput. This is done when each node reports the range of throughput that their link can handle in addition to the currently available throughput.

2. Latency: The latency is the amount of time a packet takes to travel from one point in the network to another. The latency object is calculated by calculating the estimated actual latency. This is done when each node report the range of latency that they allow in addition to the latency they are suffering based on the power consumption.

3. The Link Quality Level Reliability Metric (LQL)[4]: The Link Quality Level (LQL) object is used to quantify the link reliability using a discrete value, from 0 to 7, where 0 indicates that the link quality level is unknown and 1 reports the highest link quality level. The LQL can be used either as a metric or a constraint. When used as a metric, the LQL metric can only be recorded. For example, the DAG Metric object may request all traversed nodes to record the LQL of their incoming link into the LQL object. Each node can then use the LQL record to select its parent based on some user defined rules.

4. The ETX Reliability Object: The ETX metric is the number of transmissions a node expects to make to a destination in order to successfully deliver a packet. In contrast with the LQL routing metric, the ETX provides a discrete value (which may not be an integer) computed according to the formula below:

$$ETX = \frac{1}{PRR_{down} \times PRRup}$$

and where PRR is the Packet Reception Rate.

$$PRR = \frac{\text{Number of Received Packets}}{\text{Number of Sent Packets}}$$

and ETX is the Expected Transmission Count.

2.4.3 RPL Objective Functions

An Objective Function defines how a RPL node selects the optimised path within a RPL instance based on the routing metrics and constraints. It provides specific optimisation criteria like minimise hop count, path ETX, Latency etc. RPL forms Directed Acyclic Graph (DAGs) based on the objective function. The OF guides RPL in selection of the preferred parents and candidate parents. It is also used by RPL to compute the ranks of a node. All upward traffic is forwarded via the preferred parent. The ETX metric of a wireless link is the expected number of transmissions required to successfully transmit a packet on the link. Objective Function ETX uses ETX metric while computing the shortest path.

The Objective Function (OF) is identified by an Objective Code Point (OCP) within the DIO Configuration option. An OF defines how nodes translate one or more metrics and constraints, which are themselves defined in [66], into a value called Rank, which approximates the node's distance from a DODAG root in term of the number of hops it needs to reach it. An OF also defines how nodes select parents. When a new DIO is received, the OF that corresponds to the Objective Code Point (OCP) in the DIO is triggered with the content of the DIO. OCP is an identifier assigned by the Internet assigned Numbers Authority (IANA). Two OCP values are assigned, one for OF0 given identifier OCP 0 and the other for the Minimum Rank with Hysteresis Objective Function (MRHOF) given the identifier OCP 1. It is worth noting that OF0 and MRHOF are the only two Objective Functions that are fully defined by IETF. ETX is still a draft however it is widely used. Two other draft Objective Functions that are not used as much and are proven not to be effective are Load Balancing Objective Function (LBOF) and Traffic Aware Objective Function (TAOF).

In this section, the objective functions overview is shown with how each of them format the Destination Advertisement Object (DAO) message with values relevant to the OF and the decision of the preferred parent.

Objective Function Zero

The metrics and constraints objects discussed above in Section 2.4.2 are used, if selected in the DAG Metric Container to select the preferred parent. Each of those individually can be used to determine the path for a node to the root. However when multiple DAG Metric Containers are used, those metrics are grouped together in a Objective Function.

An OF0 implementation first computes a new variable called step of rank (SR). This variable is associated with a given parent from relevant link properties and metrics as explained below.

The SR is used to compute the amount by which to increase the rank along a particular link. It first starts by making sure the node is a candidate preferred parent (received DIO message) by making sure the link is valid in terms of connectivity and suitability. After this, the node makes sure that the candidate node has acceptable node attribute (power, energy,cpu, memory, battery) to be able to act as a preferred parent. If all those criteria are fulfilled, the node selects the candidate as a preferred parent and changes the value of its rank in the RPL DAO message by increasing the rank it received in the DIO of the candidate by 1.

The variable rank increase RI is represented in units expressed by the variable M, which defaults to the fixed constant that is defined in [4] as the default minimum hop rank increase DRI = 256.

The SR is then computed for that link by multiplying by the rank factor Rf and then possibly stretched by a term Sr that is less than or equal to the configured stretch of rank. The resulting RI is added to the Rank of preferred parent R(P) to obtain that of this node as below:

$$R(N) = R(P) + RI$$

where

$$RI = (Rf \times SR + Sr) \times M$$

Minimum Rank With Hysteresis Objective Function (MRHOF)

The Routing Protocol for Low-Power and Lossy Networks (RPL) is a distance vector IPv6 routing protocol designed for LLN networks. RPL is designed for networks which comprise thousands of nodes where the majority of the nodes have very constrained energy and/or channel capacity. To conserve precious resources, a routing protocol must generate control traffic sparingly [67]. However, this is at odds with the need to quickly propagate any new routing information to resolve routing inconsistencies quickly.

RPL organises its topology in a Directed Acyclic Graph (DAG). An RPL DAG must have at least one RPL root and a Destination Oriented DAG (DODAG) is constructed for each root. The root acts as a sink for the topology by storing all routes to all nodes in the DODAG in the routing table. The root may also act as a border router for the DODAG to allow nodes that belong to different DODAGs to communicate [4].

For a DODAG to be constructed, the root will need first to broadcast a DODAG Information Object (DIO) message, discussed in detail in Section 2.4.1, to all its neighbours. This DIO message will propagate through the network. Each node that receives a DIO message will consider the sender node a preferred parent to reach the root node until it receives another DIO message with better metrics to reach the root from another node [4].

The DIO message contains the DAG Metric Container option that is used to report metrics along the DODAG. The DAG metric Container may contain one specific metric or various numbers of metrics and constraints as chosen by the implementer [4]. Should multiple metrics and/or constraints be present in the DAG Metric Container, their use to determine the "best" path can be defined by an Objective Function (OF).

The Objective Function (OF) is identified by an Objective Code Point (OCP) within the DIO Configuration option. An OF defines how nodes translate one or more metrics and constraints, which are themselves defined in [66], into a value called Rank, which approximates the node's distance from a DODAG root in term of the number of hops it needs to reach it. An OF also defines how nodes select parents. When a new DIO is received, the OF that corresponds to the Objective Code Point (OCP) in the DIO is triggered with the content of the DIO. For example, OF0 explained in Section 2.4.3, is identified by OCP 0 by the Internet assigned Numbers Authority (IANA). The Minimum Rank with Hysteresis Objective Function (MRHOF) explained in Section 2.4.3, is the other Objective Function defined by IANA and given the identifier OCP 1.

Several Objective Functions were designed in order to fulfil specific tasks. A Destination Advertisement Object (DAO) message, for each node receiving the DIO message, will be sent to the candidate node (DIO message origin) with values relevant to the OF and the decision of the preferred parent.

This Objective Function describes the Minimum Rank with Hysteresis Objective Function (MRHOF) [68], an Objective Function that selects routes that minimise a metric, while using hysteresis to reduce lagging in response to small metric changes. First, it finds the minimum cost path, i.e., path with the minimum Rank. Second, it switches to that minimum Rank path only if it is shorter (in terms of path cost) than the current path by at least a given threshold. This second mechanism is called “hysteresis”. MRHOF works with additive metrics along a route, and the metrics it uses are determined by the metrics that the RPL Destination Information Object (DIO) messages advertise.

MRHOF uses current minimum path cost for the cost of the path from a node through its preferred parent to the root computed at the last parent selection. It also uses the following parameters

- MAX LINK METRIC : Maximum allowed value for the selected link metric for each link on the path.
- MAX PATH COST : Maximum allowed value for the path metric of a selected path.
- PARENT SWITCH THRESHOLD : The difference between the cost of the path through the preferred parent and the minimum cost path in order to trigger the selection of a new preferred parent.

- PARENT SET SIZE : The number of candidate parents including the preferred parent, in the parent set.
- ALLOW FLOATING ROOT : If set to 1, allows a node to become a floating root. A node MAY declare itself as a Floating root, and hence have no preferred parent, depending on system configuration.

On top of that, the calculation of the ETX metric is given constant selected metrics based on [69]. The metrics are:

- MAX LINK METRIC : Disallow links with greater than 4 Expected Transmission Counts on the selected path (Set to 512).
- MAX PATH COST : Disallow paths with greater than 256 Expected Transmission Counts (Set to 32768).
- PARENT SWITCH THRESHOLD : Switch to a new path only if it is expected to require at least 1.5 fewer transmissions than the current path (Set to 192).
- PARENT SET SIZE : If the preferred parent is not available, two candidate parents are still available without triggering a new round of route discovery (Set to 3).
- ALLOW FLOATING ROOT : Do not allow a node to become a floating root (Set to 0). If FR is 0 and no neighbours are discovered, the node does not have a preferred parent and must set the minimum path cost to PS.

Expected Transmission Count Objective Function

The Expected Transmission Count ETX metric discuss is based on the number of expected transmissions required to successfully transmit and acknowledge a packet on a wireless link. The ETX metric is commonly used in wireless routing to distinguish between paths that require a large number of packet transmissions from those that require a smaller number of packet transmissions for successful packet delivery and acknowledgement however RPL uses this metric to establish preferred parent based on the value of the ETX metric of the link as defined in [66] and in [70] and make it available for route selection. This is called ETX Objective Function (ETX).

In ETX, ETX metric allows RPL to find a minimum-ETX path from the nodes to a root in the DAG instance. This is the minimum ETX path between a node and the DAG root is the path (among other paths between the source and the destination) that requires the least number of packet transmissions per packet delivery to the DAG root. Thus, minimum-ETX paths are generally also the most energy-efficient paths in the network.

The ETX uses the ETX metric to find the path to be used to deliver packets in a DAG instance with the minimum number of transmission required by using the the ETX link metric to compute an ETX path metric based on the ETX link metric of each hop and choosing paths with smallest path ETX.

At first, the root node set the parameters to identify the smallest ETX path for each node:

- *min_path_etx*: A variable that determines the ETX path metric of the path from a node through its preferred parent to the root computed at the last parent selection.
- $MIN\_ETX\_PATH\_CONST$: A constant that defines the maximum ETX value that can be considered for a node to be considered for parent selection.

Each other node in the DAG (non root) computes the ETX path metric for a path to the root through each candidate neighbour by using the two parameters explained below:

- $ETX\_Neighbor\_Metric$: A variable that identifies the ETX metric for the link to a candidate neighbour
- MIN_PATH_ETX: A variable that assigns a value for each neighbour and the minimum ETX path advertised by that neighbour.

A node computes the ETX path metric for the path by comparing all the MIN_PATH_ETX received for each candidate neighbour. If a neighbour ETX metric cannot be computed, it is set to infinity to avoid selecting it and potentially having high ETX paths.

A node SHOULD compute the ETX Path metric for the path through each candidate neighbour reachable through all interfaces. If a node cannot compute the ETX path metric for the path through a candidate neighbour, the node MUST NOT make that candidate neighbor its preferred parent.

If the ETX metric of the link to a neighbour is not available, the ETX Path metric for the path through that neighbour SHOULD be set to INFINITY. This metric value will prevent this path from being considered for path selection, hence avoiding potentially high ETX paths.

The ETX Path metric corresponding to a neighbour MUST be re-computed each time the ETX metric of the link to the candidate neighbour is updated or if the a node receives a new MIN_PATH_ETX advertisement from the candidate neighbour.

After computing the ETX path metric for all candidate neighbours reachable for the current DAG instance, a node selects the preferred parent. The selection process is based on the condition that the ETX path metric corresponding to that neighbour is smaller than the ETX path metric of all the other neighbours.

Once the preferred parent is selected, the node sets its MIN_PATH_ETX variable to ETX path metric of the preferred parent. The vale of this variable is then carried in the metric container whenever DIO messages are sent.

Load Balancing Objective Function

Load Balancing Objective Function LBOF adds Child Node Count (CNC) as a metric, and uses it to select paths in a way that maintains a balanced number of children per preferred parent in the DODAG [71]. This will balance the traffic between the nodes, resulting in lower power consumption (hence longer network lifetime), a lower possibility of bottlenecks, and better delivery rate. An evaluation for this OF was carried in [72] with a comparison to OF0 and MRHOF, and it shows that LBOF provides longer network lifetime (by 16-40%) and better delivery rate (by 10-15%). However, with larger networks the LBOF seems to consume more energy due to parents churn. For this reason LBOF is considered out of context of this research.

Traffic Aware Objective Function

Traffic Aware Objective Function (TAOF) uses a combination of EXT and Packet Transmission Rate (PTR) as routing metrics, and uses it to select paths with less traffic towards the root and is defined in [73]. Authors in [74] defines TAOF which balances the traffic load that each node processes in order to ensure node lifetime maximization. They alter the DIO message format, introduced a new RPL metric, named Traffic Rate and used a new parent selection algorithm. The results in [74] show that TAOF achieves enhanced performance in terms of Packet Delivery Ratio (PDR) and that it builds more stable networks with fewer parent changes. However, it doesn't cope well with a dynamic network as it will increase the packet delivery ratio if the number of hops to reach the border gateway increases. For this reason TAOF is considered out of context of this research.

2.5 Security

Security is a major issue in the roadmap as explained in [75] to implementing the Internet of things mainly because it is not possible to directly apply existing Internet-centric security mechanisms due to the intrinsic features of WSN (e.g. the capabilities of the nodes, the bandwidth of the wireless channel) .

The purpose of those readings was to understand the standards and protocols that are becoming the driving force for securing a large network of sensors and small devices that will form the Internet of Things. This security involves securing the key establishment process and the routing discovery and establishment process.

Like any other network, the primary goals of securing the Wireless Sensor Network are the standard security goals such as confidentiality, integrity, authentication and availability.

- Confidentiality: the ability for a message to remain confidential by concealing it from a passive attacker. For a WSN, a sensor node should not reveal its data to its neighbours.

- Authentication: the ability to ensure that the message reliable by confirming and identifying the source of this message (origin). Data authentication can be achieved by verifying the identity of source through symmetric or asymmetric mechanisms

- Integrity: the ability of nodes to ensure that the message was not tampered and modified during transmission.

- Availability: the ability to use the resources and retain them for the whole duration of the communication of messages.

Other security goals such as data freshness, self-organization and secure localization are also of importance. Data freshness is the ability to ensure that the message received is the most recent one and that no newer messages were relayed. Self-organization in a network is when a node is able to self-organize and self-heal itself when it was compromised. Secure localization is the ability to locate accurately a node in a network.

Security challenges for the IoT and its integration within the IoT is studied as the challenges are tightly applicable to other relevant technologies of the IoT such as embedded systems, mobile phones and RFID. Security Threats for IoT based on the goals mentioned above are:

- Confidentiality: threats for confidentiality in IoT involves an attacker eavesdropping and overhearing critical information such as sensing data and routing information.

Based on this the adversary may cause severe damage since they can use the sensing data for many illegal purposes [14].

- Authentication: Threats for authentication in IoT involves attacks on the network that can alter the packets. It can also inject false packets. Another threat for IoT, is a general threat for wireless networks. The nature of the media and the unattended nature of wireless sensor networks make it extremely challenging to ensure authentication.

- Integrity: a malicious node present in the network can inject false data. Instability of wireless channel can cause damage or loss of data.

- Achieving a self-organizing and self-healing network in IoT is considered challenging since there is no fixed infrastructure to manage the network. This inherent feature brings another challenge as the damage resulting from an attack can be devastating.

- Localization in Wireless sensor network is essential as a compromised node can result for the attacker to manipulate data sending wrong location information by reporting false signal strengths and replaying signal.

Wireless sensor network limitations/weaknesses:

- Limited resources: for wireless sensor networks, the nodes will be limited in terms of memory, energy and processing power. Any of the security functions that will be applied on a WSN will need to take into consideration those issues as most of the available protocols and standards for encryption, decryption, data signatures, and signature verification consume memory, energy and computational power.

- Highly unreliable communication medium is another limitation for the wireless sensor networks as the nature of the communication medium can cause latency, multi-hop routing, network congestion or even conflicts such as collision. Unreliable transfers is another limitation where packets can become corrupted or even discarded which results in packet loss. This will force nodes to allocate more resources to error handling.

- On most wireless sensor networks applications, node will be left unattended and this can cause serious issues and limitation especially when nodes are exposed to physical attacks. The network is distributed thus if the design is not adequate, it can leave a network that is hard to manage, inefficient and fragile.

2.5.1 Security in RPL

Mayzaud et.al in [76] identified three different categories of attacks on RPL that can violate one or more of the security goals defined in the previous section. The first category covers nodes resources such as energy, memory and processing power. The second category includes attacks on the topology of the RPL network and the third category corresponds to attacks against the network traffic. Attacks in the first category can damage the network since all nodes are constrained and this will shorten the lifetime of these nodes. Attacks in the second category will disrupt the normal operation of the network such as how RPL network converge and the third category of attacks will violate the confidentiality and integrity of data in the RPL network.

The main focus on this research is to mitigate attacks against traffic by preventing eavesdrropping and passive sniffing. Although the first two categories of attacks are out of context of this research, we will show in Chapter 8 how encryption can prevent other attacks that fall in the other two categories such as Rank Attack and Man in the Middle attack that can disrupt the RPL network.

RPL supports message confidentiality and integrity. It is designed as such that link-layer mechanisms can be used when available and appropriate and yet in their absence, RPL can use its own mechanisms. RPL supports three security modes defined in [4].

They are Unsecured, Pre-installed and Authenticated. Unsecured refers to the security mechanism that is provided in lower layers such as link layer security. Pre-installed and authenticated modes require the use of pre-installed shared keys on all nodes prior to deploying the nodes. Both modes provide security procedures and mechanisms at the conceptual level and are concerned with authentication, access control, data confidentiality, data integrity and non repudiation. This study focuses on the Pre-installed mode as a method of securing message transmission between nodes in an RPL DAG instance. Authentication in the pre-installed mode involves the mutual authentication of the routing peers prior to exchanging route information (i.e., peer authentication) as well as ensuring that the source of the route data is from the peer (i.e., data origin authentication) [77]. The limitation of the pre-installed mode in its common form, is that it is assumed that a node wishing to join a secured network is pre-configured with a shared key for communicating with all neighbours and the RPL root. This means that once this shared key is compromised, all network leaves in the RPL DODAG are compromised.

The process of distributing the keys is out of scope for the specification of the RPL request for comment document [4]. The document further assumes that in authenticated mode , the router will dynamically install new keys once they have joined a network as a host however how the router will distribute those keys is out of context for RPL specifications and is not defined.

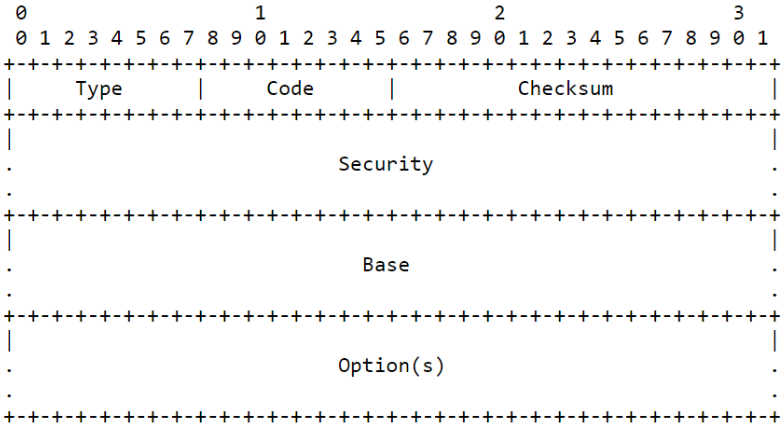
The RPL control messages incorporated in [4] the secure field in the header contents as shown in figure 2.9 below. The secure field contains several subfields as shown in Figure 2.10 and each of the subfields identify the level of security and the algorithms in use to protect RPL algorithms.

The security variants provide integrity and replay protection as well as optional confidentiality and delay protection. The optional confidentiality variant is not defined in [4] however a security algorithm is proposed to specify the encryption algorithm to be used once keys are distributed.

The main security fields shown in Figure 2.9 and Figure 2.10 are the Message Authentication Codes (MAC) and signatures provide authentication over the entire unsecured ICMPv6 RPL control message, including the Security section with all fields defined but with the ICMPv6 checksum temporarily set to zero. Encryption algorithm provides confidentiality of the secured RPL ICMPv6 message that includes the cryptographic fields (MAC, signature, etc.). In other words, the security transformation itself (e.g., the Signature and/or Algorithm in use) will detail how to incorporate the cryptographic fields into the secured packet. The Security Algorithm field specifies the encryption, MAC and the signature scheme the network uses. The cryptographic mode of operation described in [4] (Algorithm = 0) is based on CCM and the block-cipher AES-128 defined in [78]. This mode of operation is widely supported by existing implementations.

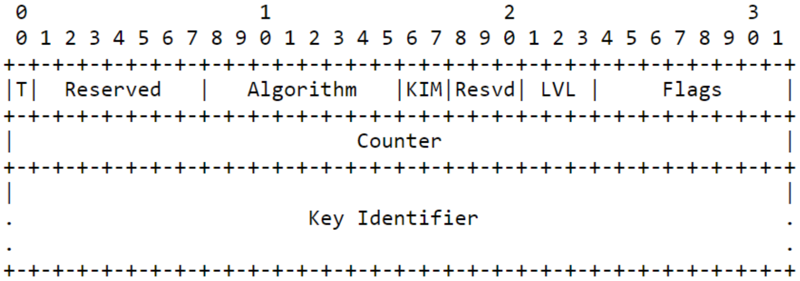
2.5.2 IoT Cryptography

The end-to-end principle argues that many functions can be implemented properly only on an end-to-end basis, such as ensuring the reliable delivery of data and the use of cryptography to provide confidentiality and message integrity. Adding a function to improve reliability on a particular link may provide some optimization, but can never ensure reliable delivery end-to-end. Similarly, security objectives that can only be met by protecting the conversation between two end-nodes are therefore best met by performing the cryptography at layer 3 or higher. There may even be security objectives that require protecting the data itself instead of the communication channel. However, this does not mean that all security objectives can be met end-to-end. In particular, achieving robust availability often requires protecting the subnetwork against attackers and more so for wireless networks. Adding a first line



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                             Security                          .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                              Base                             .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                           Option(s)                           .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.9: Secure RPL Control Message as shown in [4]. The ICMPv6 information message with a type of 155. The code identifies the type of the RPL control messages (DIO, DAO, DIS, etc..), and the checksum computation field that is computed for each security message.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|T|  Reserved   |   Algorithm   |KIM|Resvd| LVL |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Counter                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                        Key Identifier                         .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.10: Security Section as shown in [4]. The level of security of the algorithm in use are indicated in the protocol message. The algorithm field specifies the ecnryption type, the MAC and signature scheme the network uses. The counter is Time T that is a timestamp of security.

of defence at layer 2 may also increase robustness against attacks on confidentiality and integrity.

When combining encryption with authentication, some of the authenticated information may have to be sent in the clear. AES/CCM therefore encrypts a message (*m*) and authenticates that together with (possibly empty) additional authenticated data a, using a secret key K and a nonce N. A parameter L controls the number of bytes used for counting the AES blocks in the message; *m* must be shorter than 28L bytes. For IEEE 802.15.4 packets, the smallest value of L = 2 is plenty. Counter with CBC-MAC (Cipher Block Chaining Message Authentication Code) (CCM] is an authenticated encryption algorithm that provides at the same time confidentiality, authentication and integrity protection.

Even with the best link-layer security mechanisms , the data is no longer protected once it leaves the link. This makes the data vulnerable at any point that is responsible for forwarding it at the network layer, or on any link that has lesser security. Even worse, an attack on the network layer might be able to divert data onto a path that contains additional forwarding nodes controlled by the attacker. End-to-end security that protects the conversation along the entire path between two communicating nodes is therefore an important element of any robust security system, so much so, that this requirement became a banner feature in the development of IPv6 [57]

Security involves two main aspects, the Network access (authorization) and the key management during the device communication. Key management protocols can be classified according to the method the key is delivered (key transport or key agreement) and whether key exchanged are based on symmetric or asymmetric cryptography.

Symmetric techniques demand the communicating parties to possess the same key prior to message exchange. Standard online key exchange protocols involving public parameters or trusted authorities are generally avoided. Instead, as defined in [79] Key pre-distribution KPS techniques, involving the following steps are preferred: (i) Preloading of Keys into the sensors prior to deployment; (ii) Key establishment: this phase consists of (a) Shared key discovery: establishing shared keys) among the nodes and (b) Path key establishment: establishing path via other node(s) between a given pair of nodes that do not share any common key.

All of key management or key agreement schemes follow one of the three general key agreement schemes: trusted-server scheme, self-enforcing scheme, and key pre-distribution scheme. Trusted server scheme is not suitable for wireless sensor network as usually there is no centralized infrastructure in sensor networks such as a centralized entity to manage Kerberos. The self-enforcing scheme depends on symmetric cryptography such as a key agreement using a public key certificate. Limited computation and energy resources of sensor nodes often make it undesirable to use public key algorithms such as Diffie-Hellman key agreement or RSA. Many implementation and evaluation proved this to be an unrealistic scheme for WSN [52] to use Public Key Infrastructure (PKI) technology. For example, each endpoint must be able to store digital keys, run encryption and decryption algorithms and conduct sophisticated handshakes to establish secure SSL connections, etc. However, many IoT nodes like the passive RFID tags or sensors simply don't have the electrical power, storage, or processing power necessary to tackle even the simplest of PKI tasks.

The time to execute the main cryptographic operation of ECC, the scalar point multiplication has been reduced from 34 seconds in 2004 to less than 0.5 seconds in 2009. With ECC, any node can make use of digital signature schemes (ECDSA), key exchange protocols (ECDH), and public key encryption schemes (ECIES). However, PKC is still too expensive to be used by sensor nodes implementing web servers as the overhead of its software implementation (420 ms) is too high. Note that the use of other PKI primitives with extremely efficient encryption and verification is discouraged. However PKI is still too expensive to be used by sensor nodes implementing web servers, as the overhead of its software implementation (420 ms) is still too high. [80]

IPsec was considered a serious contender for securing WSN and many methods of research were involved in creating a lightweight version of IPsec to be incorporated into the 6LoWPAN architecture. Authors in [81] and [82] suggested compressing the IPsec and only looked at the authentication header part of the IPsec but suggested to use key pre-distribution for the end to end communication. Other research suggested that the IPsec is unsuitable ias t is designed for one to-one communication. However, the dominant types of communication in WSNs are Many-to-one and One-to-many. This makes such protocols unsuitable for the usage in WSNs.

Sensors can use the 6LoWPAN protocol to interact with an IPv6 network as they are powerful enough to implement symmetric key cryptography standards such as AES-128 in [83]

It was very important to understand how those networks utilize the available pre-distribution techniques such as the mostly used one, proposed by Eschenauer & Gligor in [46] to secure the Distributed Sensor Networks (DSN).

Authors in [84] modified E-G scheme by only increasing the number of keys that two random nodes share from at least 1 to at least q. It increased vulnerability in a large scale node compromise attack. They further extended this idea and developed two key pre-distribution techniques: a q-composite key pre-distribution scheme and a random pairwise keys scheme.

The q-composite key pre-distribution also uses a key pool but requires two nodes compute a pairwise key from at least q-pre-distributed keys that they share. The random pairwise keys scheme randomly picks pairs of sensor nodes and assigns each pair a unique random key.

A framework was developed in [85] to be used to improve the performance of any existing key pre-distribution scheme using polynomial pairwise key . This framework does not require any prior knowledge of sensors' expected locations, and thus greatly simplifies the deployment of sensor networks.

Authors in [80] explained that even if assumptions were made that a WSN peer is protected by its own security mechanisms such as using the link layer security of IEEE 802.15.4, the public nature of the internet will require the existence of a secure communication protocol for protecting the communication between two peers.Key establishment is a fundamental security issue in wireless sensor networks (WSN). It is the basis to establish secure communication using cryptographic technologies between sensor nodes. Due to the current resource constraints on sensors, it is infeasible to use traditional key management techniques such as public key cryptography or key pre-distribution centre based protocols. Therefore the key pre-distribution schemes are paid most attention in key management of WSN.

It is now accepted to assume that the Key management scheme for distributed sensor networks developed by Eschenauer & Gligor is a standard to use for securing wireless sensor networks. However Eschenauer & Gligor only looked at the key pre- distribution schemes proposed for WSN and ZigBee as the main purpose of their research, our objective is to implement a Key distribution mechanism for the IoT to solve the problem of exchanging key between devices connected to the IoT without compromising the nodes or the validity of the Keys because of a Man in the Middle attack using the same scheme proposed by Eschenauer & Gligor in [46]. Algorithm for the key management scheme for distributed sensor networks and how it will be used in the context of the IoT will be shown later on in this chapter in Section 2.5.4 and in Chapter 4.

2.5.3 DSN Key Pre-Distribution

In order to provide security between nodes communicating, encryption/decryption keys needs to be used for each and every communication link between devices. The main feature of key pre-distribution and how it works is referred in the context of any Ad Hoc network as a challenge. The challenge simply lies in how the keys will be distributed beforehand and how to ensure that nodes communicating in an Ad-Hoc nature share a key and thus can provide secrecy and authentication by encrypting their communication channel.

The management of key is one of the key challenges to secure networks. We list below key pre-distribution challenges when used in the context of the distributed sensor networks DSN.

- It is difficult to distribute keys and keying materials such as identifiers prior to deployment.
- Nodes in the networks are not authenticated and therefore obtaining a key does not guarantee that a node is trusted.
- Nodes in the distributed sensor networks are mostly battery operated low power devices, limited memory resources and computation power and the key pre-distribution scheme chosen needs to have low overhead to ensure that the nodes can still operate efficiently.
- The nature of the distributed sensor networks and where nodes are located means that it is difficult to know where nodes. This can potentially result in the physical capture of the nodes and they become compromised and all credentials can be exposed.
- Note all nodes are implemented at the same time, for this reason the key pre-distribution scheme needs to ensure that existing nodes in the network will work together securely with the newly added nodes.
- If node is compromised,

In addition the challenges to the key pre-distribution presented above, the Internet of Things network present on top of those challenges other challenges unique to them. The main challenge related to this research is the nature of how nodes communicate in an IoT network which prevent nodes from creating more than one node and therefore if the key distribution scheme used does not produce enough keys not all nodes will participate in the IoT network.

In sensor networks, key pre-distribution is usually combined with initial communication establishment to bootstrap a secure communication infrastructure from a collection of deployed sensor nodes. In the setting we study in this Chapter, nodes have been pre-initialized with some secret information before deployment, but only after network setup will we know the location of nodes. The node location often determines which nodes need to establish a link with which other nodes, so we cannot set up these keys before deployment. In this Chapter, we refer to the combined problem of key pre-distribution and secure communications establishment as the security bootstrapping problem, or simply the bootstrapping problem. A bootstrapping protocol must not only enable a newly deployed sensor network to initiate a secure infrastructure, but it must also allow nodes deployed at a later time to join the network securely.

This is a challenging problem due to the many limitations of sensor network hardware and software. In this Chapter, we discuss and evaluate several well-known methods of key distribution. Besides these, we present an in-depth study of random key pre-distribution, a method that has recently attracted significant research attention and we have also worked on. However, the pairwise key establishment problem is still not solved. For the basic Probabilistic and the q-composite key pre-distribution schemes, as the number of compromised nodes increases, the fraction of affected pairwise keys increases quickly. As a result, a small number of compromised nodes may affect a large fraction of pairwise keys. While the random pairwise keys scheme doesn't suffer from the above security problem and given the memory constraint, the network size is strictly limited by the desired probability that two sensors share a pairwise key and the number of neighbour nodes that a sensor can communicate with.

The interest of this research is to look at the various methods of key distribution between various devices in the context of the IoT proposed and study their feasibility.

Pre-distribution of keys can follow one of three major approaches when used in the context of the IoT as explained in [86]. The Probabilistic approach explained in Section 2.5.4, the Deterministic approach explained in Section 2.5.5 or the hybrid approach that combines both as proposed in [87], [88], [89] and [90] .

Paterson & Stinson mathematically investigated in [91] the metrics that should be used to assess the suitability of the various Probabilistic and Deterministic key pre-distribution schemes and identified them as the network size, storage requirements, network connectivity and network resilience. When using those Key pre-distributions schemes in the context of the IoT other metrics also needs to be evaluated as proposed in [92]. The metrics are scalability to identify if the scheme can support large networks, efficiency to evaluate how much storage and processing power the used scheme will use, storage complexity in term of the amount of memory required to store the security keys for large networks and processing complexity in order to computer the amount of processor cycles required to establish a key and communication complexity as in the number of messages exchanged during the key generation and distribution process. Resilience should also be considered in evaluating how resilient the network will be if a node is captured and keys need to be revoked. Finally the key connectivity metric will need to be evaluated as the number of keys will increase if the probability of two nodes to share a key is low and this will have a high impact on the other metrics.

2.5.4 Probabilistic Key Pre-Distribution

Probabilistic schemes is where the secure link establishment is conditioned by the existence of shared pre-loaded keys and Deterministic schemes which ensure total secure connectivity coverage. The idea behind the Probabilistic scheme was proposed first by Eschenauer & Gligor in [46]. A Random key pre-distribution (RKP) where each node is pre-loaded with a key ring of m keys randomly selected from a large pool. After the deployment step, each node exchanges with each of its neighbours the key identifiers that it maintains in order to identify the common keys. If two neighbours share at least one key, they establish a secure link and compute their session secret key which is one of the common keys. Otherwise, they should determine secure paths composed by successive secure links.

Traditional key exchange and key distribution protocols based on infrastructure using trusted third parties are impractical for large scale distributed sensor networks. There is no key distribution at the moment implemented on DSN other than key pre-distribution. However the key pre-distribution offers two inadequate solutions: Single mission key solution is inadequate because if one sensor node was compromised, this would lead to the compromise of all the DSN since selective key revocation is impossible upon sensor capture detection

The other solution is pair wise private sharing of keys avoids compromise of the whole DSN since it allows selective key revocation. However, it requires pre-distribution and storage of n-1 keys in each sensor. This will mean that each node will require a large amount of memory to store the keys if for example a DSN contains 1 000 nodes. In total there will be $n(n-1)/2$ keys per DSN. It will also render the communication between the devices complex and resources draining.

Eschenauer's & Gligor's approach was to propose a single key pre-distribution scheme that requires memory storage for only a few tens to a couple of hundred keys, and yet has similar security and superior operational properties when compared to those of the pair wise private key sharing scheme.

Their scheme relies on Probabilistic key sharing among the nodes of a random graph and uses a simple shared key discovery protocol for key distribution, revocation and node re-keying.

This research will look in Chapter 4 at how the Probabilistic key pre-distribution scheme can be applied in the context of the Internet of Things networks to allow keys to be distributed among nodes in the network so that only RPL nodes that share a pair-wise key can join the RPL DODAG.

2.5.5 Deterministic Key Pre-Distribution

Deterministic schemes ensure that each node is able to establish a pair-wise key with each of its neighbours. To guarantee determinism, Localized Encryption and Authentication Protocol (!LEAP) explained in [93], makes use of a common transitory key that is pre-loaded into all nodes prior to deployment. The transitory key is used to generate session keys between neighbouring nodes before being removed.

The scheme suggested by [47] divides the solution into three phases. In the first phase, each node attempts to discover which nodes are within its neighbourhood and to verify their identities. For this, each node will commit to each identity discovered in its neighbourhood and perform the fingerprinted mutual authentication protocol FMAP protocol with each neighbour it is supposed to share a key with. The FMAP protocol assumes that each node that is pre-loaded with the fingerprint of every other node. Each node that joins the network broadcast a simple HELLO message containing its fingerprint and its key list. Every node that receive this message can verify the fingerprint in order to confirm uniqueness. If a similar fingerprint exists, the node is not allowed to join. At the end of the first phase, each node will have a list of all its neighbours including identity and fingerprint and will have verified the identity with neighbours that it shares key with. At this stage, nodes have not decided whether to accept this identity or not. Each node will overhear all FMAP protocol messages in order to decide whether it accepts its identity or not. In Phase 1, each node n_i has now established a path with all direct neighbours that it was able to identify their identity of the form $n_i \rightarrow n_j$.

In the second phase and since a node has already identified direct neighbours that it shares a key with, the next step is to identify if a path can be established further beyond neighbours by using them as hops - That is the neighbours that exist outside of n's neighbourhood in the form of $n_i \rightarrow n_j \rightarrow n_k$. Verifying a node that is not a direct neighbour is more difficult as FMAP protocol cannot be imitated on nodes that are not neighbours (Those nodes cannot respond to HELLO messages from neighbours of neighbours). For this n_i will have to rely on the trust issued by each of its direct neighbours to their corresponding neighbours. However it cannot assume that the process of identifying of its neighbours n_j assumption about the identity is correct. For this it applies a voting process in which if the majority of nodes that are direct neighbours identify n_k as their direct neighbours then it assumes that n_k is an honest node. Since n_k is trusted by the majority, it is now considered as a trusted device by n_i and thus a 2 hop path is established.

In Phase 1, each n_i learns paths of the form $n_i \rightarrow n_j$, and in Phase 2 each n_i learns paths of the form $n_i \rightarrow n_j \rightarrow n_k$. Just as nodes informed their neighbours of the results of Phase 1 so that the information could be utilized to construct 2-hop paths, each node broadcasts the results of Phase 2 so that nodes of their neighbourhood learn which 3-hop paths exist. More specifically, each n_j will broadcast all paths it has discovered of the form $n_j \rightarrow n_k \rightarrow n_l$. This way, in phase 3 each node increases its knowledge of the network by one hop by relying on the nodes that were verified during phase 1 and 3 of the protocol. In phase 3, n_i is not voting for the majority to decide whether to trust n_l and has to trust that n_j already has chosen n_l as it gained majority.

This research will look in Chapter 5 at how the Deterministic key pre-distribution scheme can be applied in the context of the Internet of Things networks to allow keys to be distributed among nodes in the network so that only RPL nodes that share a pair-wise key can join the RPL DODAG.

2.5.6 Threats Attacks Trees

Internet of Things networks are subject to several threats as discussed in Section 1.1.2 and identified which threats can be mitigated by using encrypted communication between nodes in the network.

In this section we will look at the different threats that can be carried by malicious actors and the attack surfaces that can be exploited in order to compromise the network. We categorized the threats identified in Section 1.1.2 into two different type of attacks. The first category of attacks explained in Section 2.5.6 assumes that the malicious actor is exploiting the link of nodes that are sending data in plain text and on the encryption algorithm used to protect the link. The second category investigated in Section 2.5.6 shows how a malicious actor can attempt to exploit the routing formation or the routing table.

IoT Confidentiality and Integrity of data Attack Tree

The threat of having an insecure communication between IoT devices is now more tangible than a conventional threat for any other type of networks on the Internet. Plain text communication makes it easier for attackers to tamper with data as well. In another scenario where an attacker tampers with the communication between an IoT device sending regular measurement of a valve in a factory for another machine to switch off for example at a critical level and modifies the temperature data. This can potentially be disastrous for a factory and might even lead to loss of life. We present in Figure 2.11 below the attack tree that results in violation of confidentiality, integrity or availability of the RPL DODAG.

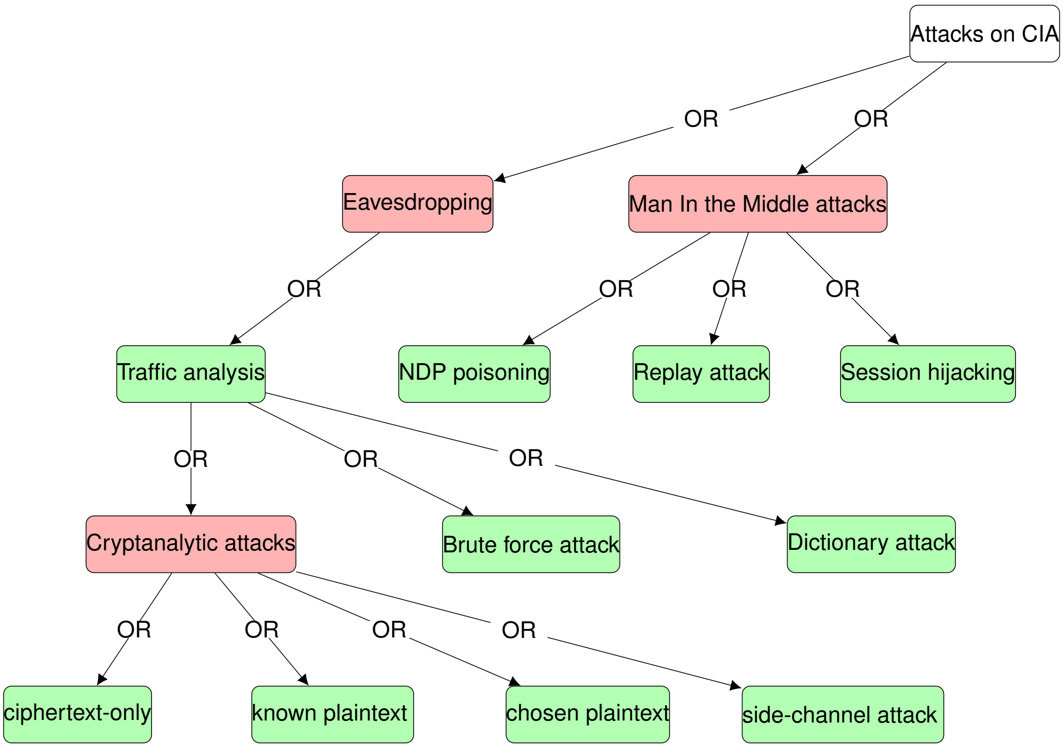


Figure 2.11: Attack tree representation of all the attacks on the confidentiality, Integrity and Accountability that an Internet of Things network is vulnerable when all communications are sent in plain text.

Man in the Middle (MiTM) attacks will allow a malicious actor to eavesdrop into the communication and sniff the data transmitted between nodes. This will reveal both the data information exchanged between nodes and the control messages between nodes such as routing table formation. Since MiTM is most of the time used to allow further attacks such as session replay where the attacker stores messages exchanged between nodes in order to replay them later on. This will potentially lead to repudiation of data as there will be no method to identify and validate if the data sent is correct and the malicious actor can tamper with the data.

The proposed solution can protect some of the attacks presented in Figure 2.11. Traffic analysis can partly be prevented as the traffic is encrypted and the payload (data) is not sent in the clear text. Having the data sent in clear text will violate the confidentiality of the data. This will also lead to violation of the integrity if further attacks are carried out such as Man in the middle attacks that can easily be done if the traffic is sent in clear text. The different cryptanalytic attacks presented depends on the encryption algorithm used.

IoT Routing Table and Formation Attack Tree

In Figure 2.12 we present how the RPL routing table or the RPL topology maintenance can be attacked. We note that they all rely on the presence of one or more malicious nodes in the network. A malicious node can disrupt the RPL DODAG formation and results in one of the attacks explained in Section 1.1.2, however, if the nodes communicate using the proposed solution and form secure links they can prevented.

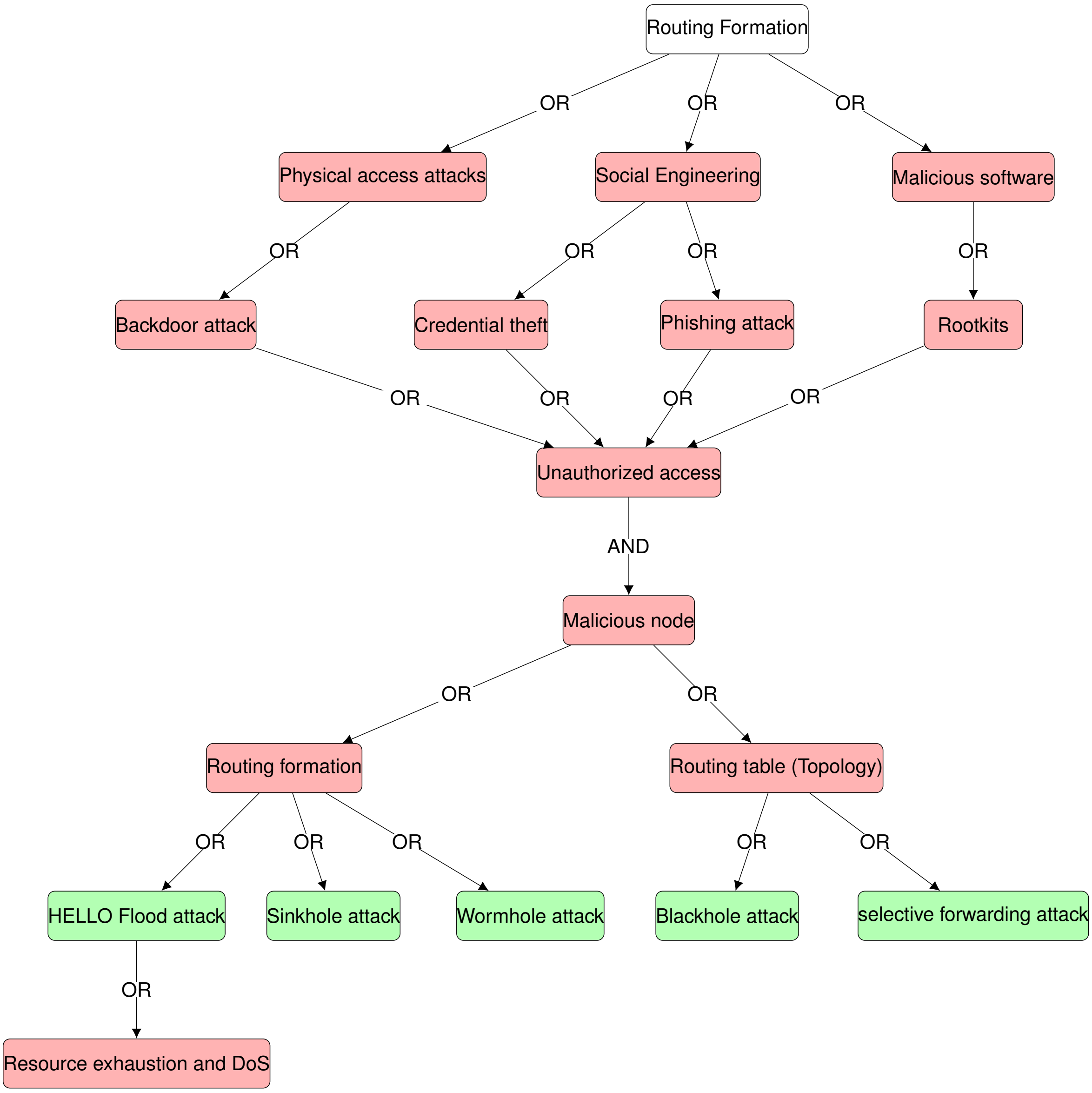


Figure 2.12: Attack tree representation of all the attacks on the routing formation when all the routing control messages are sent in plain text.

2.6 Summary

In this Chapter, we first defined the differences between the Wireless Sensor Networks WSN, the Distributed Sensor Networks DSN, and the Internet of Things IoT. The differences are mainly related to the link availability between nodes in the network since nodes between DSN and WSN are between each node and all its neighbours in comparison with the IoT networks where each node form a link only with one preferred neighbour based on certain variables.

We then introduced the IoT 6LoWPAN concept that defines how the Internet Protocol can be used in the context of the Internet of Things and researched the routing power for loss networks RPL and explained how it works and the various objective functions that can be used and the security measures that are incorporated within it.

We finally discussed the threats and vulnerabilities that IoT nodes and networks are vulnerable to and researched different key distribution schemes that are available and how each of them is used in order to identify in later Chapters which one of is more suitable to use in the context of the Internet of Things.

Chapter 3

Testbed Design and Methodology

This Chapter proposes two test-beds designed to investigate how suitable key pre-distribution schemes are in the context of the Internet of Things and what were the necessary modifications that were made to improve the performance of one of the key pre-distribution schemes.

The idea of using a simulator to simulate the experiments of this research is of a viable solution in research specially when it comes to using Internet of Things networks. An Internet of Things network is usually of a large scale and contains thousands of nodes. For this reason and for practicality, it was essential to use Contiki to simulate such large networks. The simulation testbed is described in 3.2. We use the simulator in Chapters 4, 5 and 6.

To ensure real world implementation yields to the same result, we evaluate the performance of all schemes explored in this document using Zolertia devices in Chapter 7. The practical simulation testbed is described in 3.3. We identify variables and look on the reasoning behind the choices used for the parameters chosen. We also look at various random generators and how keys and identifiers were generated.

3.1 Research Methodology

This thesis is a combination of quantitative and qualitative research, it will involve first determining the acceptable parameters needed to provide secure IoT using various key pre-distributed schemes. This will need to include an extensive literature review of the various key pre distributed schemes available and the various parameters used to evaluate their performance. After this those parameters will need to be evaluated in a simulated environment to ensure their validity.

The second phase, which includes build and experiment methodology used to develop a test-bed in order to test the selected pre-distribution schemes in the context of the IoT. Developing the test-bed should be studied carefully as the number of devices used in the test-bed and their interoperable functions should be wisely chosen in order for the test bed developed to be as close to a real life scenario. This is necessary in order to demonstrate that it is possible to use the KPS scheme on the Internet of Things.

The third phase continues in the context of experimental methodology in term of collecting records of simulation to compare and evaluate the results of the performance of the test bed using the parameters chosen at the beginning and the acceptable quality of service for the internet of things.

3.2 Simulation Test-Bed

A simulation environment allows researchers to implement a large scale network that is not usually feasible financially and logistically in real life if done for research only.

The ability to embed your own code, be that a software or a specific protocol on a large number of devices deployed together with the same characteristics is another reason why many researchers opt to carry experiments in a simulated environment rather than a hardware and physical deployment. Contiki is an open source introduced in [94] is a highly portable multitasking operating system, in which the 6LoWPAN has been implemented. In Contiki , only several *K* Bytes of code and a few hundred bytes of memory are required to provide a multitasking environment and built-in TCP/IP support. This makes it especially suitable for memory constrained embedded platforms .

Contiki OS is an Operating system for the Internet of Things that contains several simulation environments built up into several tools to produce a closest to possible real life scenario. It helps facilitate the deployment of large networks by ensuring that applications designed for low power device will work well in a simulated environment and debug program before being pushed into real environment. It was developed at the Swedish Institute of Computer Sciences by Adam Dunkels et al. Contiki is a highly portable OS and it has already been ported to several platforms running on different types of processors. It uses an event-driven programming model to handle concurrency and all processes share on stack. This allows devices to save memory, which is an important factor in low power devices. Contiki uses Protothreads to do this as it provides conditional and unconditional block wait and use for the various states of hardware components in the device. Several Internet protocols were ported to Contiki and it supports both IPv4 and IPv6 stack implementation and various low power wireless standards such as the 6LoWPAN stack and the rime stack.

Cooja is a Java-based simulator designed for simulating sensor networks running the Contiki sensor network operating system [95]. The simulator is implemented in Java but allows sensor nodes software to be written in C.

Tunslip is another tool in Contiki that we use to bridge IP traffic between a host and another network element, typically a border router, over a serial line. Tunslip creates a virtual network interface (tunnel) on the host side and uses SLIP (serial line internet protocol) to encapsulate and pass IP traffic to and from the other side of the serial line.[96]

3.3 Hardware Test-Bed

After completing the simulation of various experiments, we move to experimenting with the same variables in a hardware environment. The practical environment that we implement is composed of 15 Zolertia node devices called Z1 shown in figure 3.1a and an Onion border gateway router shown in Fig3.1b. We first flash the firmware of the devices with the same simulation firmware we were using in the simulation environment. This is to ensure similar data. The firmware we flash contains two rings that were picked randomly from a pool. This was done for each device. Once the firmware was pushed we then use firmware of the border gateway router to ensure it uses RPL with the pre defined settings we modified for all experiments such as the Objective Function metrics and objects. For each experiment, we push a new firmware with different variables as defined by the experiment objectives.

The Z1 node hardware is a Wireless Sensor Network node that is equipped with a an MSP430F2617 low power microcontroller as explained in [97]. It features a built in 8KB RAM and 92KB flash memory.

3.4 Experimental Design Overview

The Experiments will be looking at finding and comparing the ring size required in a selected network for all nodes to share a key and thus be able to communicate in a secure way. In the following sections we introduce the experiment testbed both in a simulated environment and the hardware environment.

3.5 Experimental Procedure & Variables

- Each simulation will be run using a specific number of keys/IDs in the pool.
- For each specific number of keys/IDs in the pool, a size of two rings will be defined, one for the keys ring and another for the identifiers ring.

- The simulation will contain several experiments that will increment the number of nodes.
- Each experiment will be run 5 times.

The structure of each experiment is based on Pool Size P and the ring size changing once for each simulation. For each simulation, the pool size is only generated once and is used for all runs regardless of the size of the Network. The ring size will depend on the size of the pool. The only variable that will changes in different experiments inside a simulation is the number of nodes in the network.

The variables can be divided into three categories

- Control Variables : Variables that will stay constant for the remainder of the experiments explained in Section 3.5.
- Independent Variable: variables that will change during the experiments to reflect the desired network metrics of a specific simulation explained in Section3.5.
- Dependent Variables: Variables that are obtained from experiment simulations explained in Section3.5.

For the purpose of the experiment, we decided to use the values for all of the experiment's simulation. The control variables were chosen as assumptions based on the intended experiment and the results wanted.

- The key length (**klength**) of 64 bits which is more than enough for the number of nodes we plan to run in this simulation. This is a combination of a 40 bits key and 24 bits Initialization vector.
- The ID length (**ilength**) of 32 bits which is more than enough for the number of nodes we plan to run in this simulation. The number of bits in ID was chosen to be smaller because of memory constraints in the Internet of things devices. The other reason is that exchanging those bytes is not revealing anything as there is no connection between keys and IDs is exchanged. Anyone trying to intercept the messages will not be able to make the relation between the ID exchanged and the key that will be used.
- Transmitting range of 50 meters for small networks and 25 meters for large networks: Both ranges are common for the Internet of things nodes deployed on Contiki. The zolertia node is capable of using both 50 meters range on short range frequency of 865

MHz of and 20 km on 2.4 GHz frequency as discussed in the technical documentation of Zolertia zoul in [98]. The Nodes Openmotes, seedeye, sky and wismote are also emulated in Contiki as seen in [99] with a range of 50 meters.

- X and Y maximum coordinates: At the moment, we are running the simulation on an area of 250 meters by 250 meters. This applies to all simulations regardless of the number of nodes. 250*250 is a reasonable size for a large environment such as a university. Therefore, the maximum X and Y coordinate is 250. This of course can be changed later, if needed.
- Number of runs: At first in a small environment, the experiment was replicated 20 times with the same variables similarly to the sampling size estimation in [100]. We then calculated the average value after removing the highest and lowest values for accuracy. This method was repeated again with 15 runs and 10 runs until we identified that the experiments do not return any significant difference between 20, 15, 10 and 5 runs. For this reason and for the efficiency of the experiments it was determined that 5 runs is more than enough to obtain good consistent results. We tried to run it with 8 runs and the numbers did not change at all.

The independent variables calculated from the Equation 3.5.1 proposed in [46] are shown below.

- Pool size (**P**): Two pools are being generated in each simulation, one for keys and the other for IDs and both have the same size. The pool size is an important factor that will have a huge impact on the probability of shared keys between nodes. The pools size we run simulations for are: (100, 250, 500, 750,1 000) and (2 500). A pool of 15 keys and another for 15 identifiers are generated for the practical experiments only.
- Number of nodes (**N**): The number of nodes is related directly to the pool size and for each simulation we decided to evaluate the performance of each pool on various network sizes starting from 15 nodes to 2 500 nodes. The number of nodes in the network are shown in Table3.1 below.
  - Based on the mathematics in [46] using a pool larger than a 100 keys or identifiers in a small network yields to very large rings. This is not realistic and it was decided that experiments should only run in a realistic environment; Network sizes of 15, 25, and 50 not used in experiments where the pool was larger than 100.

- Ring size (RS) : This is another important factor that will have a big impact on the probability of finding shared keys between two nodes. Both shared keys and shared identifiers will lead to a secure communication as each identifier represents one key. For practicality we use (RS) when we need to represent both **KRING** and **IRING** and they are referred to respectively when specifically needed. The choice of Rings size used in the simulations done is based on the equation in [46] shown below in Equation 3.5.1 where $p' = 0.5$, **P** is the size of the pool. k is the ring size value that represents RS. We obtained the ring size shown below in Table 3.1.

$$p' = 1 - \frac{(1 - \frac{k}{P})^{(2(P-k+\frac{1}{2})}}{(1 - \frac{2k}{P})^{(P-2k+\frac{1}{2})}} \tag{3.5.1}$$

Table 3.1: Independent variables that will be used for the simulations. The pool size incrementing from 100 to 2 500 keys and identifiers and the starting ring size values calculated from the Probabilistic scheme ring size values for DSN. The network size will change from 100 to the size of the pool except in smaller pool where the network is incremented from 15 to the pool size.

Pool size	Ring size	Network size					
100	8	15	15	25	50	100	-
250	13	100	250	-	-	-	-
500	18	100	250	500	-	-	-
750	22	100	250	500	750	-	-
1000	25	100	250	500	750	1 000	-
2500	41	100	250	500	750	1 000	2 500

Dependent variables are the variables obtained after running the simulation. Those results will be obtained from the routing table and from comparing the rings between the nodes that formed the dags. Below we will find an explanation of all the variables that we will be measuring and collecting in each experiment.

3.5.1 DAGS Number

The number of in a network is related to the number of nodes that are available in the RPL DODAG.

The following explanation of how the DAGs are formed and the number of DAGs calculated is taken from [101] and is only included here for completion. The formulation of DAGs in RPL is modelled as a simple, strongly connected directed graph $G = (N,E)$, where N is the set of nodes and E is the set of links. In addition, if there exists an edge from vertex to vertex , there also exists an edge in the reverse direction (from vertex m to vertex n); that is, the graph is symmetric. A set of candidate paths $p = 1,2,,,,.p$ is provided, each path represent a sequence of C_p directed edges. The identifier of the destination (root) node is denoted by r. Neighbours of node m or r are indexed by n. The explanation of how the DAGs are formed and the number of DAGs calculated is shown below and explained in more detail in [101]. The following variables are present in the formulation:

1. $y_{mn} = 1$ if the (m,n)edge is included in the DODAG0 otherwise;
2. $Y_p = 1$ if candidate path p is included in the DODAG0;
3. U_m is an integer variable representing the number of paths from a node to the root in the DODAG but only counting the paths included in the set inserted into the formulation. No other paths available in the DODAG are counted by this variable.

The problem is formulated as follows:

$$\mathrm{lexmax}\{U^{m}(Y) = [U_{m1}(Y), U_{m2}(Y), ...U_{m|N|-1}(Y)]\} \tag{3.5.2}$$

where

$$U_{m1}(Y) \leq U_{m2}(Y) \leq ...... \leq U_{m|N|-1}(Y)]\} \tag{3.5.3}$$

Apart from the root, each node has at least one outgoing edge:

$$\sum_{0} Y_{mn} \geq 1 \quad \text{for} \quad \text{each} \quad m \subset N \quad , \quad m \neq r \tag{3.5.4}$$

All edges incoming to the root are included in the DODAG:

$$Y_{mr} = 1 \quad \text{for} \quad \text{each} \quad \text{neighbour} \quad n \subset N \tag{3.5.5}$$

At the same time, all edges outgoing from the root are excluded from the DODAG:

$$Y_{mr} = 0 \quad \text{for} \quad \text{each} \quad \text{neighbour} \quad n \subset N \tag{3.5.6}$$

Of each edge pair (m,n) and (n,m) not adjacent to the root, not more than one should be selected:

$$\begin{gathered} Y_{mn} + Y_{nm} \leq 1 \quad for \quad each \quad pair \quad of \quad neighbouring \\ nodes \quad m,n \subset N \quad , \quad m \neq r \quad n \neq r \end{gathered} \tag{3.5.7}$$

All possible cycles must be eliminated. Cycle of length 1 do not exist in the network graph, as it is assumed to be simple. Cycles of length 2 are already eliminated by constraints (3.7.6). Hence, it is required to eliminate cycles of length $K = 3,4,......|N|$. For this purpose, the following constraints can be formulated:

$$\begin{gathered} Y_{k1k2} + Y_{k2k3+......+Y_{kKk2}} \leq K - 1 \\ for \quad each \quad interconnected \quad nodes \quad k_1,k_2k_3,k_4.....k_K \subset N \end{gathered} \tag{3.5.8}$$

If any of the edge variables belonging to a path is equal to 0, then the variable is also equal to 0, edges incoming to the root are always included in the DODAG:

$$\begin{gathered} Y_p \leq Y_{mn} \quad for \quad each \quad pair \quad of \quad consecutive \\ nodes \quad m,n \subset N \quad on \quad the \quad path \quad p \quad except \quad r(n \neq r) \end{gathered} \tag{3.5.9}$$

The last expression represents the number of paths from node m to the root:

$$\begin{gathered} U_m = \sum_p Y_p \quad for \quad each \quad m \neq r \quad and \quad paths \quad p \\ originating \quad at \quad node \quad n \end{gathered} \tag{3.5.10}$$

3.5.2 Shared Keys (SK)

The Shared Keys (SK) is an essential variable for this research. This variable is related to the number of nodes in a network that share a key. Sharing a key allow the two nodes to communicate securely.

For example, suppose that two nodes (A) and (B) have two key rings sets (k_A) and (k_B) where k of each is a node that contains RS number of keys as defined in Section 3.5. If (k_A) ∩ (k_B) then one or more shared keys exist. SK would be equal to 100% assuming that only nodes (A) and (B) exist in the network.

3.5.3 Hop Count Average

In a Directed Acyclic graph, the hop count is simply related to the number of hops a packet needs to go through before it reaches the root node. The rank of a node is directly related to the number of hops. For example, for a child node (Rank 3) to send or receive packets to any other node through the root node its packets will need to jump two hops in the form of child → parent → root as shown in Figure 3.2 below.

When looking at hop count, we can calculate the rank of a node in a Directed Acyclic graph . The rank of the node is a direct relationship of how far it is from the root node. The rank of a node is calculated using the below relation where the rank comparison is the comparison between two different ranks as in $OldRank == NewRank$ where $OldRank$ is the rank a node has and $NewRank$ is the rank obtained after a new $DAGRank()$ was calculated.

$$\text{RankComparison}(\text{DAGRank}()) \tag{3.5.11}$$

Where the rank is a fixed-point number that is determined by the $MinHopRankIncrease$ variable, $MinHopRankIncrease$ is the minimum hop rank increased between a node and any of its DODAG parents. This variable is essential when using Objective Function zero (OF0) explained in Section 2.4.3. It determines paths of all nodes by selecting paths that have smallest number of hops.

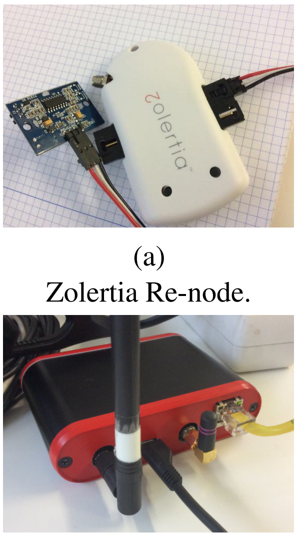
The rank quantity in RPL is of 16 bits length. When the Objective function computes the rank, it uses the rank quantity length of two nodes and compares them to computer $DAGRank()$ where one can be a parent and the other a child of this parent. The integer portion of the Rank is computed by the $DAGRank()$ macro where $floor(x)$ is the function that evaluates to the greatest integer less than or equal to x using the align below.

$$\text{DAGRank}(\text{rank}) = \text{floor}(\frac{\text{rank}}{\text{MinHopRankIncrease}}) \tag{3.5.12}$$

An example of how the value of the $DAGRank(rank)$ is shown in the RPL request to comment document RFC6550 [4] where if a 16-bit Rank quantity is decimal 27, and the MinHopRankIncrease is decimal 16, then $DAGRank(27) = floor(1.6875) = 1$. The integer part of the Rank is 1 and the fractional part is $\frac{11}{16}$.

By using the calculations above to determine the DAGRank of a specific node $DAGRank(node)$. Each node in the network will have this calculation in the form of $DAGRank(node.rank)$, where node.rank is the Rank value as maintained by the node.

Once the rank is calculated for two nodes, we can then determine their location in the DODAG. Suppose we have two nodes A and B. Both nodes have the $DAGRank$ calculated as in $DAGRank(A)$ and $DAGRank(B)$ then:



(a)
Zolertia Re-node.

(b)
Onion Border gateway router.

Figure 3.1: Devices for the practical testbed environment. The physical environment comprises 15 Zolertia renode and and the border gateway router. The 15 devices (nodes) creates the 6LoWPAN internet of Things network and the one of the node acts as a root node in the network to translate all the IoT traffic to the router. The router sends the data to the Internet for analysis and processing.

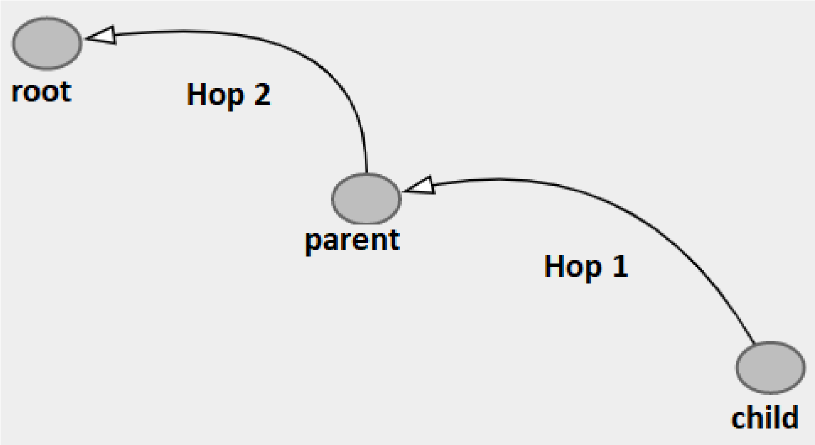


Figure 3.2: An example of how Hops count from a child to root passing through a parent is calculated. The rank of a node is directly related to the number of hops.

- Node A has a Rank less than the Rank of a Node B if $DAGRank(A) < DAGRank(B)$. In this case the position of B is closer to the root of the DODAG than A.
- Node A has a Rank equal to the Rank of a Node B if $DAGRank(A) = DAGRank(B)$. In this case the position of both A and B in relation to the root of the DODAG is the same.
- Node A has a Rank greater than the Rank of a Node B if $DAGRank(A) > DAGRank(B)$. In this case the position of B is further to the root of the DODAG than A.

For the whole DODAG and for all experiments, two values of the Hop Count explained below are calculated.

1. The average value of the number of hops for all nodes in a DODAG in the first five minute of the experiment is called Initial Average Hop Count (IAHC).
2. The average value of the number of hops for all node in a DODAG once RPL Converge called is Converged Average Hop Count Converged Average Hop Count (CAHC).

3.5.4 Latency

Latency is used in certain Objective Functions for RPL as a metric such as when the Rank is computed using the ETX (expected transmission count) Objective Function. Nodes in this OF optimize themselves to determine parents using the latency link metric. The Latency link metric is the time it take for each node to send data to its parent node. Each node will report the latency (delay) of receiving a packet from a child node.

In this research the latency variable (LAT) shown in Equation 3.5.13 is directly related to the Converged Average Hop Count ($CAHC$). It is calculated in Equation 3.5.4 by measuring the average latency for each hop (ALN) first and multiplying this by the number of hops as in below.

$$\mathrm{LAT} = \mathrm{ALN} \times \mathrm{CAHC} \tag{3.5.13}$$

$$\overline{\mathrm{ALAT}} = \sum_{\mathrm{i}=1}^{\mathrm{x}-1} \frac{\mathrm{LAT_i}}{\mathrm{x}-1} \tag{3.5.14}$$

Where x is the number of nodes in the network that are participating in the RPL DODAG and share at least one key and $x-1$ is the number of nodes participating in the network without the root node.

3.5.5 Power Consumption

Power consumption estimation in Contiki is computed using Energest module. The Energest module is a lightweight software provided by Contiki to estimate energy estimation for resource-constrained IoT devices. It does this by tracking the time various hardware components such as the radio are turned on, and by knowing the power consumption of the component it is possible to estimate the energy consumption. By knowing the duration each hardware component in a node was being used, Energest module can then estimate the power consumption of that component based on how long it was used. Contiki initialize various states that contributed to the total power consumption of a node. The different states shown in Figure 3.3 are CPU usage time, Low Power Mode (LPM) time, Interrupt Request (IRQ) or no radio, Transmit (Tx) time and Receiver (Rx) time. Each of those states will consume power differently from one device to another and when the state is on or off.

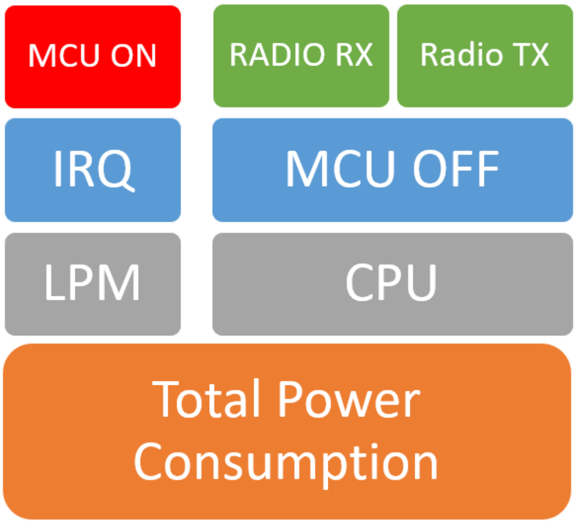


Figure 3.3: Power consumption distribution showing how the power is distributed between the MCU, IRQ, LPM, CPU and radio Rx and Tx power consumption.

For example, a Zolertia Re-node node operates with a voltage of 3V. Looking at power consumption calculation in PowerTrace Contiki in [98] datasheet, we can see that each state consumes different power as shown below:

- If MCU is on for device:20mA
- If MCU is idle for device (IRQ): 0.55mA
- If MCU is on for radio RX: 22mA

- If MCU is on for radio TX: 20mA
- If MCU is on for CPU: 40mA

Once the voltage for each state is identified we can calculate its power consumption. For this we use both PowerTrace and Energest with Cooja.

To estimate the energy consumption, we start with selecting the number of tickers per seconds for rtimer for Zolertia where in Cooja $RTIMER\_SECOND = 32768ms$. This will allow us to calculate all power consumption estimates of each state. We can then calculate the total energy consumption as in Equation 3.5.15 and duty cycle in Equation 3.5.16.

$$\text{TotalPower} = \frac{\text{Energest\_value} * \text{Current} * \text{voltage}}{\text{RTIMER\_SECOND} * \text{Runtime}} \tag{3.5.15}$$

$$\text{DutyCycle} = \frac{\text{Energest\_TX} + \text{Energest\_TX}}{\text{Energest\_CPU} + \text{Energest\_LPM}} \tag{3.5.16}$$

3.5.6 RPL Control Messages Number

Packet Delivery Ratio (PDR) computes the ratio between the total number of packets received at the root node compared with the total packets sent from all nodes. The higher the PDR is the better the performance of the routing protocol RPL.

The number of packets that each node sends or receives is measured and the average of each of them is then calculated.

The average number of packets sent per node NTx then computed as $\overline{NTx} = \sum_{i=1}^{x} \frac{NT_{x_i}}{x}$ where x is the number of nodes in the network that are participating in the RPL DODAG and share at least one key and the average number of packets received per node NRx then computed as $\overline{NRx} = \sum_{i=1}^{x} \frac{NR_{x_i}}{x}$, where x is the number of nodes in the network that are participating in the RPL DODAG and share at least one DAG.

3.5.7 Time to Converge

This variable measures the time it takes for the RPL routing table to converge. The RPL routing table converges when the DAG stops changing for a period.

3.5.8 CPU Usage

This calculation measures the CPU usage in the first 5 mins in comparison with 24 hours. The CPU usage measures how much percentage of the CPU maximum processing power is being used. The same variable is measured twice, first when RPL is still converging (after 5 minutes) and the second time after 24 hours.

The average initial CPU usage $ICPU$ in the first 5 minutes for all nodes is then computed as : $\overline{ICPU} = \sum_{i=1}^{x} \frac{ICPU_i}{x}$, where x is the number of nodes in the network that are participating in the RPL DODAG and share at least one key.

The average converged CPU usage $CCPU$ in the first 5 minutes for all nodes is then computed as : $\overline{CCPU} = \sum_{i=1}^{x} \frac{CCPU_i}{x}$. where x is the number of nodes in the network that are participating in the RPL DODAG and share at least one key.

3.5.9 Number Of Neighbours

The number of neighbours for one node is the number of nodes that fulfil the three conditions listed below in relation to one node:

1. It needs to be within range of the node.
2. It needs to share a key with this node.
3. It needs to be in the routing table RPL.

The average number of neighbours is then computed as : $\overline{N} = \frac{\sum_x (N_x)}{x}$ where x is the number of nodes in the network that are participating in the RPL DODAG and share at least one key.

3.6 Simulation Experimental Setup

The Simulation experiment is designed to work on the Contiki Operating System. The Simulation experiment is composed of five parts shown below and an explanation of how they interact with each other is explained in a flowchart in Figure 3.4.

1. Header file containing declaration of variables for different experiments.
2. C program to generate Keep pool,ID pool,Key rings and ID rings.
3. Cooja simulator composes of border router and nodes.
4. Tunslip tool to create a bridge between border router and all other nodes
5. Perl program to analyse logs logged by individual nodes after simulation

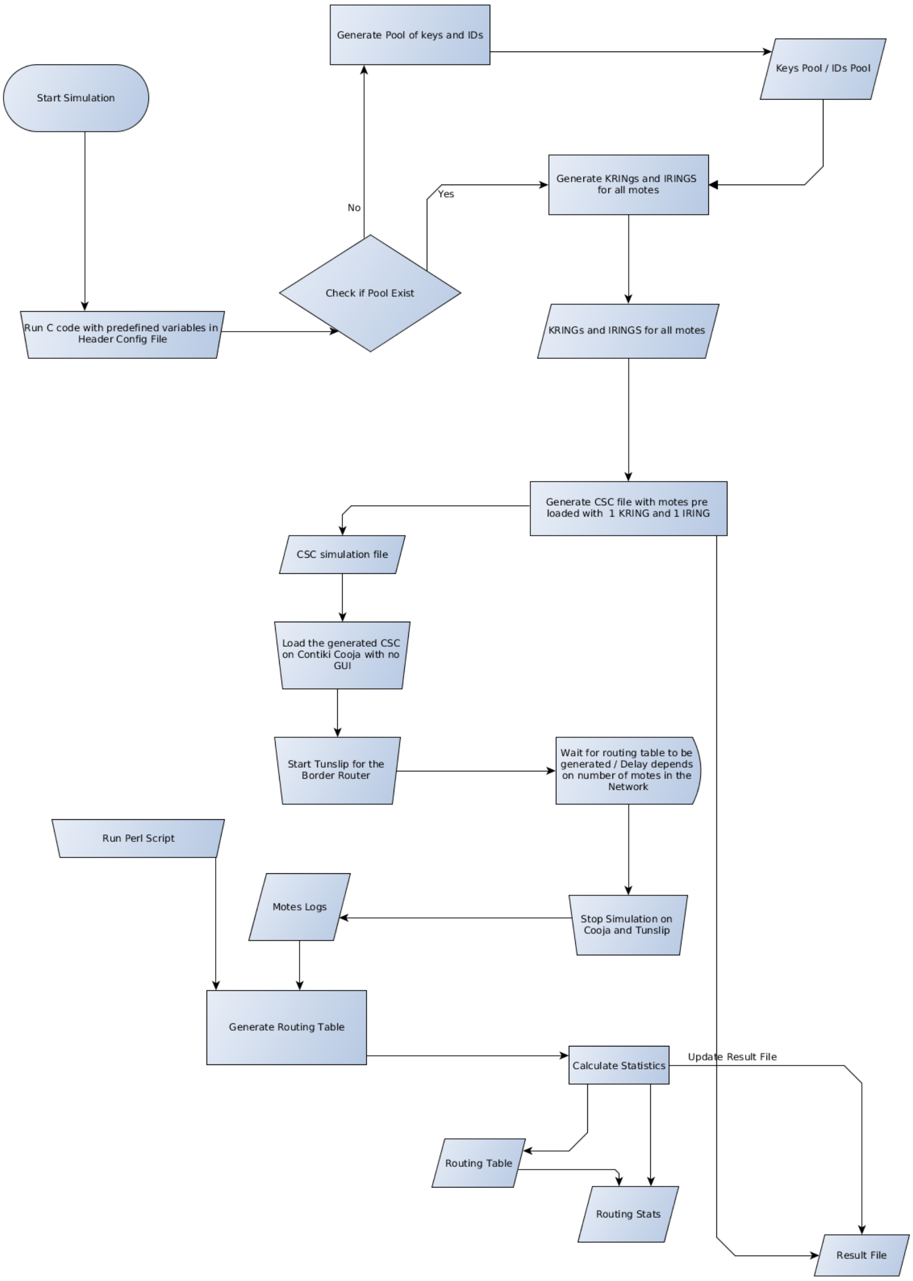

Generate Pool of keys and IDs
Keys Pool / IDs Pool
Start Simulation
Generate KRINgs and IRINGS for all motes
Yes
No
Check if Pool Exist
Run C code with predefined variables in Header Config File
KRINGs and IRINGS for all motes
Generate CSC file with motes pre loaded with 1 KRING and 1 IRING
CSC simulation file
Load the generated CSC on Contiki Cooja with no GUI
Start Tunslip for the Border Router
Wait for routing table to be generated / Delay depends on number of motes in the Network
Run Perl Script
Motes Logs
Stop Simulation on Cooja and Tunslip
Generate Routing Table
Calculate Statistics
Update Result File
Routing Table
Routing Stats
Result File

Figure 3.4: Simulation structure showing how all component of the simulation are used- Starting from the generation of the keys and identifiers using the independent variables set in the configuration of the simulation program shown in Code Listing 3.2. The Cooja simulation file is generated from the configuration file. All dependent variables are then collected from the generated logs of the experiment.

In the next section, we present the header file used for each experiment. In this file we set up the environment that will be tested. This includes the control variables and the independent variables.

3.6.1 Header File

This is the first step in the Simulation to setup the experiment variables as shown in Code Listing 3.1 below. As it stands, all experiments are running with independent variables changed for each experiment. When running one experiment all variables are fixed except the "generate pool" =0 variable which is changed to 1 only at the beginning of each experiment with a specific number of keys in the pool to generate one pool to be used for one experiment. The "generate pool is variable is changed back to 0. Once the experiment is run for 5 times the generate pool variable is then changed back to 1 for a new experiments set.

```
#define POOL_SIZE 250
#define KEY_LENGTH  64
#define ID_LENGTH 32
#define IRING_LENGTH 13
#define KRING_LENGTH 13
#define NUM_NODES 250
#define max_y_dimension 500
#define limit_y_dimension 500
#define X_LIMIT 250
#define Y_LIMIT 250
#define TRANSMITTING_RANGE 50
#define int generate_pool=1;
```

Listing 3.1 Header file - Experiment setup of the control and the Independent variables.

C Program

The program is mainly used to generate keys and IDs that form the pool. Three pseudo random number generator algorithms were used in the program. They are explained in more detail later in Section 3.8

The C program diagram shown below in Figure 3.5 describes what the program does from the beginning when it is executed to the end when each node in the network is loaded with a KRING and an IRING in the network.

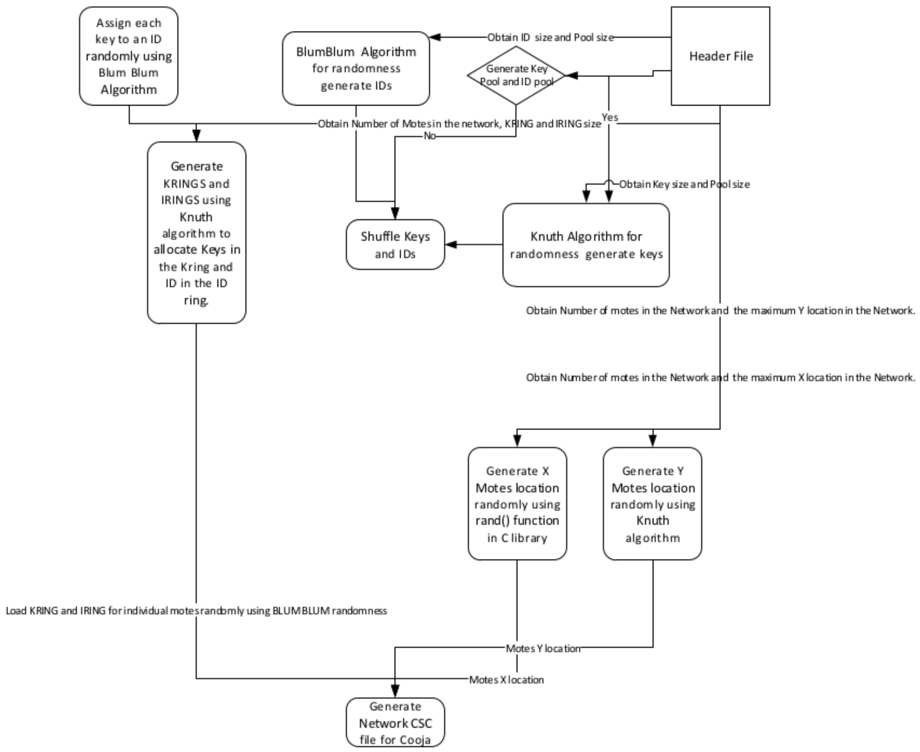

Assign each key to an ID randomly using Blum Blum Algorithm
BlumBlum Algorithm for randomness generate IDs
Obtain ID size and Pool size
Header File
Generate Key Pool and ID pool
Yes
No
Obtain Number of Motes in the network, KRING and IRING size
Generate KRINGS and IRINGS using Knuth algorithm to allocate Keys in the Kring and ID in the ID ring.
Obtain Key size and Pool size
Shuffle Keys and IDs
Knuth Algorithm for randomness generate keys
Obtain Number of motes in the Network and the maximum Y location in the Network.
Obtain Number of motes in the Network and the maximum X location in the Network.
Generate X Motes location randomly using rand() function in C library
Generate Y Motes location randomly using Knuth algorithm
Load KRING and IRING for individual motes randomly using BLUMBLUM randomness
Motes Y location
Motes X location
Generate Network CSC file for Cooja

Figure 3.5: C file Diagram showing how the pool keys and identifiers are generated and the rings randomly created and distributed to be allocated to nodes in the csc file. This is relevant for both the Probabilistic and Deterministic key pre-distribution schemes.

Perl Program

Perl script is executed after the simulation finishes. The script mainly has three tasks:

1. Read all log files generated by individual nodes and build the routing table. This is done by looking at "The preferred parent statement" for each node.

2. Compare KRINGs between nodes that makes a leaf in the RPL DAG. A leaf in the RPL DAG means two nodes sharing a direct route that is part of the RPL routing table.

3. Return statistics of the number of nodes in the network, number of nodes in the routing table and the number of nodes in the routing table that shares a key. The results are written in the routing stats file.

4. Collect dependent variable values from PowerTrace, Energest and Objective Function metrics from RPL logs.

5. Timestamp all events and group them in two categories, either events happened on or before the 5 minutes duration in an experiment run or events happened after the 5 minutes run and before 24 hours has passed.

Below we show two snippets from the Perl code. The first code shown in Code Listing 3.2 shows how the root is defined when reading the logs. The second code shown in Code Listing 3.2 show how the routing table is decided. The program searches the logs for the last message from the client to the router declaring which node is their preferred parent before the simulation ends.

```
#IP address for the root
$ip_id{'fe80::c30c:0:0:1'} = 1;
#debug printing of ids matching to IP addresses
if ($debug_print) {Winter
foreach $elem (sort {$a cmp $b} keys %ip_id) {
print "Value of element:$elem is:", $ip_id{$elem}, "\n";
```

Listing 3.2 Perl file looking for the root log

```
for ($i=2; $i <= $num_nodes; $i++) {
$filename = "logs\\log_" . $i . ".txt";
if (open ( $fh_in , '<:encoding(UTF-8)',$filename)) {
while (<$fh_in>) {
chomp $_;
if ($_ =~ m/^.* RPL: The preferred parent is (.*) \(.*\)$/) {
$preferred_parent=$1;
$preferred_parent_table{$i} = $preferred_parent;
undef $preferred_parent;
```

Listing 3.3 Perl file checking for routes in each node

3.6.2 Experiment Process

First, when executing the program, the program looks at the header file to decide if a pool of keys and a pool of Identifiers needs to be generated or if they already exist. If the pools do not exist, the program using three variables from the header file (pool size, key size, ID size) generates keys and IDs as big as defined in term of their length and the number based on the pool size. Those keys and IDs are generated randomly using two different Pseudo Random Number Generators (Knuth algorithm for Keys and Blum Blum Shub algorithm for IDs)

Second, the program needs to assign each key to an Identifier (ID). It does this in two steps, first using the shuffle function from the C library, it then shuffles all keys and IDs so that if any order exists, it will be removed. The second step uses Blum Blum Shub to allocate a key number and an ID number to be assigned to each other. Each key number chosen with its ID number will be removed from the pool so that no key/ID is assigned to the same ID/Key (No replacement).

Thirdly, the program now, depending on how many nodes exist in the network will select keys randomly from the pool and store them in a KRING for each node. This results in several KRINGs equal to the number of nodes in the network. Each node has its own KRING. The method of allocating Keys for a KRING is random using the Knuth algorithm. Two restrictions exist, first the random number cannot be greater than the number of keys in the pool and second when a number is chosen, it will not be taken from the pool but placed back (with replacement) in the pool. The IRING is formed after the KRING is formed by just choosing the IDs that maps with the keys in the KRING.

Now that a key pool and ID pool exist and each key is assigned its own ID, the nodes in the network are now generated. Nodes have two variables that determines their locations **x** and **y**. Both **x** and **y** are chosen randomly using the rand() function for coordinate **x** and Knuth algorithm for coordinate **y**. The only restriction to their location is the network area specified in the header file. For example in the header code above in 3.1, **x** and **y** are set to 500 limit. Each of the nodes in the network now has a random location with (x,y) coordinates that was chosen randomly.

The last and final step that the program does when executed is to generate a csc file (cooja simulation) that will be used in the next step of the experiment. Each node in the network is allocated a KRING and an IRING. The allocation is based on the node ID in the cooja csc file.

Final note, the generation of key pool and ID pool is only done if generate pool is 1. if it is 0, it means that the pools exist from previous simulation in the same experiment.

3.7 Fixed Network Experiment Configuration

In this section we setup a fixed experiment environment with pre loaded keys with known values and fixed location in the simulation environment. We first generate a pool of 100 keys of 8 bits each. The distribution of the pool is based on the Knuth algorithm pseudorandom generator (PRNG) briefly explained in Section 3.8. Once the keys are generated, we create 15 random rings of 8 keys with replacements where the key is chosen and replaced back in the pool. This is to ensure that the rings will have a shared key. The rings and how they are distributed is listed in Appendix C Section 3.2.

The rings and their keys for each node result in shared keys between nodes in an average of 46% of shared keys exist in all the nodes. This matches with the results that are obtained in [46]. In this experiment we also ensure a fixed location for each node in the simulated environment as shown in Figure 3.6. We ensure that the simulated environment runs in a high density environment of $50 * 50$ meters and where each node has a transmitter/receiver range of 50 meters.

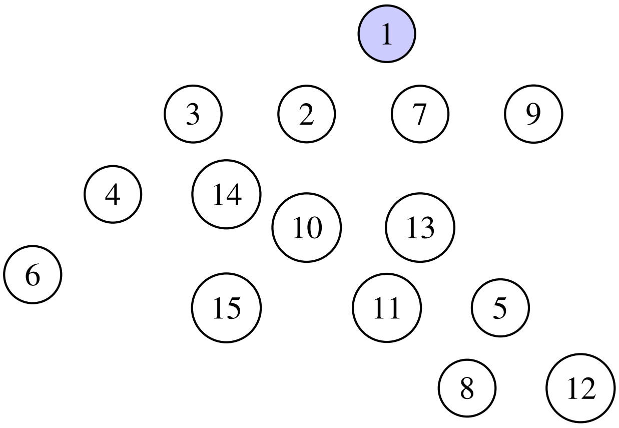


Figure 3.6: Fixed network experiment topology- key and identifiers rings are not generated randomly and are distributed using a fixed structure shown in Appendix C.

3.8 IDs and Keys Generation

One essential assumption when looking at identifiers and keys generations is that the generation of keys and identifiers is random and that probability sampling is the method used to form the various keys and identifiers rings.

3.9 Parameter Choice Problems

The original intention for this study was to reach the value given in the example of [46] when deciding on the number of nodes in the network for each simulation. The number of nodes in the example was 100 000 nodes. However the experiment faced several obstacles that made it not possible to reach this number. Each of those obstacles is listed below with what attempt was made to overcome it.

1. **Problem**: Contiki RPL was behaving as expected when the number of nodes in the network was small. As soon as the number of nodes became larger than 500 nodes,

the latency became very big and therefore many DIO messages were lost. This is only applicable to simulations using Contiki.
Attempt to solve it: The reason that many messages were lost was that the Radio Duty Cycle was running. This meant that nodes will go into sleep mode regularly. Since the number of nodes in the network was very large, most of the nodes were waking up at intervals where no messages were sent to them and thus they were not receiving any messages for a long time. We modified the code so that nodes stay awake even if the radio duty cycle is on. This is not a realistic approach but it was done for the sake of the simulation and since energy consumption for each node was out of scope and concern for this simulation. This solved the problem and we were able to resume simulation for up to 2 500 nodes.

2. **Problem**: Once we reached 1 000 nodes, a new problem related to the size of the network came up. We were running out of memory as the routing table was getting very large and there was no space for all hops to be stored.
Attempt to solve it: We disabled downward routes storage in RPL as it was not needed for this experiment. This solved the problem temporarily until we reached 5 000 nodes. We could not generate a routing table for a network this size. One attempt was to leave the simulation running for seven weeks but it was not successful.

It is also worth noting that no one to our knowledge was able to create a simulation of this size. All attempts for simulating 6LoWPAN were for networks smaller than 5 000 nodes. Many developers in the mailing list of Contiki, asked us what steps we followed in our research to achieve the 2 500 nodes.

Chapter 4

Probabilistic Key Pre distribution scheme

In this chapter, we investigate how the key distributed scheme proposed in [46] performs in an Internet of Things environment. We first experiment with the key distribution by looking at how many nodes will share a key in a network when the the ring size used is based on the calculation of the ring size using Equation 3.5.1. We run this experiment using both ETX and OF0 objective functions for RPL in Section 4.2. If the ring size used did not achieve the desired full connectivity of the network, we increase the rings size gradually until we reach full connectivity for both ETX and OF0 in Section 4.3. Once we reach full connectivity we evaluate the impact of the increased or decreased ring size on the performance of both the link metrics of the DODAG and the nodes metrics for all nodes in the network.

1. Identify if network connectivity is fully achieved by checking if all nodes are in the RPL and communicate securely in Section 4.2
2. if not, identify the required ring size for each pool that is needed to achieve the full connectivity by increasing or decreasing the number of keys and identifiers in the rings until full connectivity is reached in Section 4.3

A key management scheme for distributed sensor networks DSN proposed by Eschenauer & Gligor in [46] tends to be the standard scheme for the DSN. Traditional key exchange and key distribution protocols based on infrastructure using trusted third parties are impractical for large scale DSNs. There is no key distribution at the moment implemented on DSN other than key pre distribution. However the key pre-distribution offers two inadequate solutions: Single mission key solution is inadequate because if one sensor node was compromised, this will lead to the compromise of all the DSN since selective key revocation is impossible upon sensor capture detection The other solution, pair wise private sharing of keys avoids compromise of the whole DSN since it allows selective key revocation. However it requires pre distribution and storage of n-1 keys in each sensor. This will mean that each node will require a large amount of memory to store the keys if for example a DSN contains a 1 000 node. In total there will be n(n-1)/2 keys per DSN. It will also render the communication between the devices complex and resources draining.

Eschenauer & Gligor's approach was to propose a single key pre-distribution scheme that requires memory storage for only a few tens to a couple of hundred keys and yet has similar security and superior operational properties when compared to those of the pair wise private key sharing scheme. The scheme relies on Probabilistic key sharing among the nodes of a random graph and uses a simple shared key discovery protocol for key distribution, revocation and node re-keying.

4.1 How does It Work

First and prior to DSN deployment, a key ring of keys is distributed to each sensor node and each key ring consisting of randomly chosen k keys from a large pool of P keys which is generated offline. Even if two nodes do not share a key because keys are generated at random, the pair of nodes can use the path of an existing pair wise path to exchange keys and establish a direct link. This brings us to the main outcome of this approach, full shared key connectivity offered by pair wise private key sharing between every two node becomes unnecessary. For example: to establish almost certain shared key connectivity for a 10 000 nodes network a key ring of only 250 keys have to be pre distributed to every sensor node where the keys were drawn out of a pool of 100 000 keys leaving a substantial number for DSN growth. The approach of this scheme was divided into three different phases:

1. Key Pre Distribution
2. Shared Key discovery
3. Path key establishment.

A pseudo-code algorithm was written in order to implement the first and the second phases of the pre distributed key scheme. The third phase of the scheme is not part of the implementation since this is where the routing and end to end communication is made. The RPL protocol is responsible for creating a path between nodes and creating a routing table for all nodes in the network. The RPL routing protocol is already implemented in Cooja. The first phase of the scheme "key pre-distribution" pseudo code algorithm is shown in Appendix A, Algorithm A.1.

In phase 1, the Key pre distribution phase, five steps occur offline:

1. First a large pool of P keys is generated and their key identifiers
2. Random drawing of K keys out of P without replacement to establish the key ring of a sensor
3. Loading the key ring into the memory of each sensor.
4. Saving of the key identifiers of a key ring and associated sensor identifier into a trusted controller node.
5. Loading the i-th controller node with the key shared with that node.

This key pre distribution mechanism ensure that any two nodes share at least a key with a chosen probability of 0.5. Only 75 keys drawn out of 10 000 keys need to be on any key ring. The second phase is the shared key discovery phase, which takes place during DSN initialization where every node discovers if it shares a key with each of its neighbours within wireless communication range as explained in [**?**]. Two methods exist here; the first one which is the simplest way is that each node broadcasts in clear text the list of identifiers of the keys on their ring. This approach does not give an attacker anything new as the attacker can capture a node by decrypting communication. The other method which exists is to hide key sharing patterns among nodes from an adversary thereby establishing private shared key discovery. For every key on a key ring, a node will broadcast a list (α) where Eki $((\alpha))$, i = 1,....,k... where (α) is the challenge. The decryption with the proper key will reveal the challenge and establish a shared key with the broadcasting node. The third phase in which the shared key phase nodes establish the topology of the sensor array of the DSN. A link exists if both sensors share a key. The routing protocol is responsible for initializing the route discovery and creating a routing table for each node. RPL will be used in this case.

The algorithm of how the key distribution is used in the context of the Internet of Things is shown in Appendix A Section A.1. The second phase shown in Algorithm A.2 below, is the shared key discovery phase which takes place during DSN initialization where very node discovers its neighbours in a wireless communication range. Two methods exist here, the first one which is the simplest way is that each node broadcast in clear text the list of identifiers of the keys on their ring. This approach does not give an attacker anything new as the attacker can capture a node by decrypting communication. The other method is to hide key sharing patterns among nodes from an adversary thereby establishing private shared key discovery.

4.1.1 Key Pre-Distributed Scheme Features

The key pre-distribution scheme has several different characteristics that deals with either nodes that are compromised or if the key lifetime has expired and re-keying needs process needs to start. In this section we look at the features of the scheme.

- Revocation: is when a sensor node is compromised. It is essential to be able to revoke the entire key ring of that node otherwise not only the shared key is compromised but all the other in that ring. The controller broadcast a single revocation message containing a signed list of k key identifiers for the key ring to be revoked. The controller generate a signature key *ke* and unicast it to each node by encrypting it with a key Kci. Each node when it receives the unicast verify *ke* to locate the corresponding keys from its key ring. Thus some links may disappear. When a link disappear, then the affected nodes need to establish a new link by restarting the shared key discovery phase.
- Re-keying: the lifetime of a key shared usually exceed that of the nodes. In some case the lifetime of the shared key expires. Re-keying will also restart the process of shared key discovery.
- Sensor Node capture and resilience: the first method to capture a sensor node is by manipulating active sensor data inputs. This threat is hard to prevent and it may not be practical nor even possible to detect it. The only possible detection of this is by analysing the data correlation to look for a data anomaly. The second level is when all the sensor nodes are under the physical control of the attacker.

We have decided in Chapter 3 on the metrics for this phase that are suitable for the IoT implementation such as the size of the pool, the length of keys and their identifier and the number of nodes for the implementation. We will also have to decide on the size of the Key ring and its identifier ring. This will also depend on how the probability of finding a common key between two nodes.

4.2 Network Connectivity With DSN RSs

In these experiments set, we run the experiments with the RSs identified in Table 4.1 when using both objective functions ETX and OF0. We first identify if the percentage of shared keys between nodes is still around the 50% mark that was computed in [46].

We start by generating key rings and identifier rings using the RS variable computed in the DSN Equation 3.5.1 proposed in [46]. We have identified that the ring sizes needed for the various pool identified in Chapter 3 and shown below in table 4.1.

Table 4.1: Ring size for various pools using DSN calculation. The shared key percentage for the different pools show that the around 50% of nodes share keys between each other.

Original values			
P	N	RS	SK %
100	100	8	50.52
250	250	13	50.43
500	500	18	57.14
750	750	22	49.47
1000	1000	25	57.14
2500	2500	41	48.19

We observe in Figures 4.1 and 4.2 the percentage of shared keys for various pools size and the different network sizes. The percentage of shared keys for small networks is lower than the average for the rest. The lower density of 15 nodes and 25 nodes produces a network that is dispersed in such a way that most of the nodes are out of the transmitting range of the border router and thus the number of dags in the network is quite low. The average percentage of shared keys (PSK) between nodes in the DODAG becomes consistent around the 50% for larger networks. In fact, this starts to get more stable when the number of nodes becomes larger than 50 nodes.

This experiment also shows that regardless of the pool size, the percentage of shared keys for different network size is relatively the same regardless of the size of the key. This in turn can be validated when compared with the experiment results of [46] and the equation proposed where 50% shared keys achieves a full connectivity for DSN networks.The only exception is when the number of nodes is 15, this for the same reason explained above is when the network had a low density of 15 nodes. Many of those 10 nodes are out of reach of each other thus coming out of the network and are not included in the calculation.

In the next section, we investigate the rings size needed to achieve full connectivity in all experiments in an IoT environment using ETX and OF0 objective functions.

4.3 Network Connectivity With IoT RSs

In the previous Section 4.2, we obtained the number of keys and identifiers needed in each key ring and identifier ring in order to achieve full connectivity when using the key pre-distribution scheme in the content of the IoT running RPL with both OF0 and ETX. In

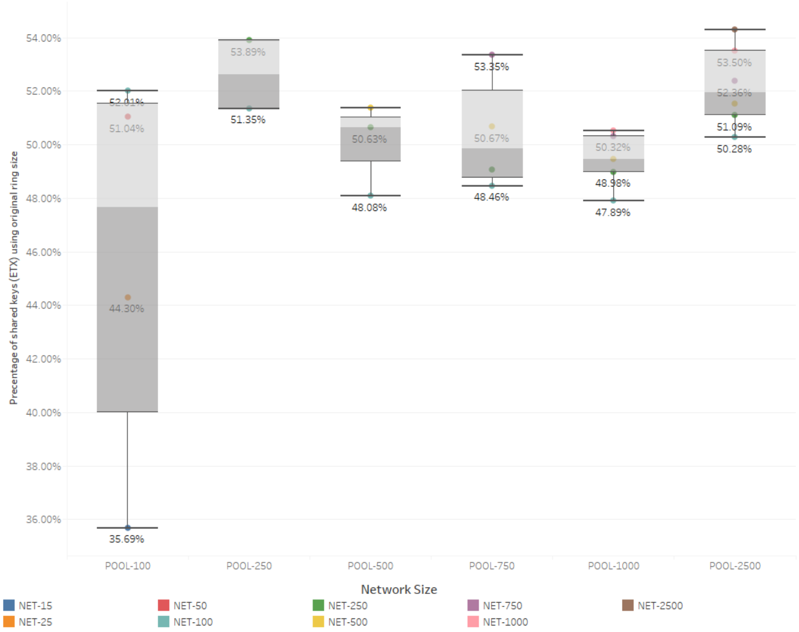


Figure 4.1: Percentage of shared keys ($PS K$) for different networks size and pools size when using ETX with DSN RS values obtained in Table 4.1. In average, only around 50% of the nodes share a key with one or more direct neighbour.

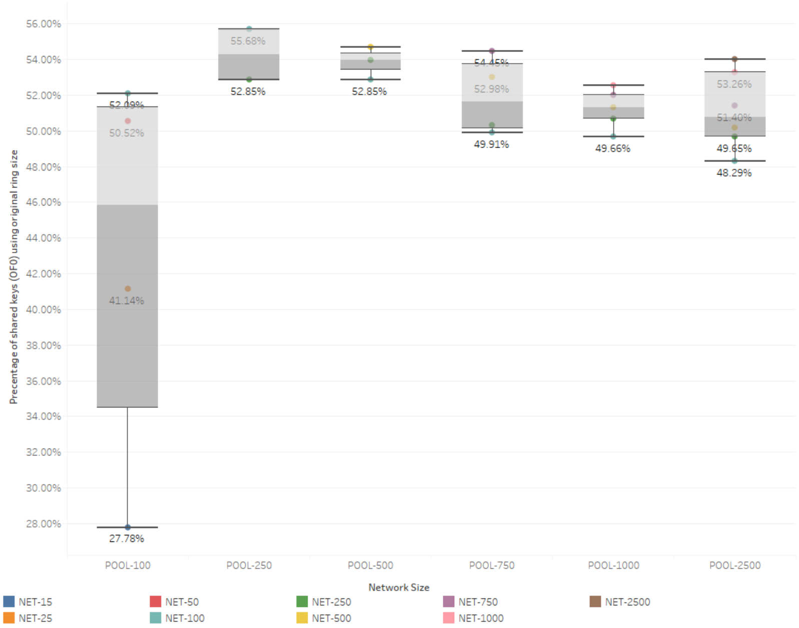


Figure 4.2: Percentage of shared keys ($PS K$) for different networks size and pools size when using OF0 with DSN RS values obtained in Table 4.1. In average, only around 50% of the nodes share a key with one or more direct neighbour.

this section we experiment increasing the rings size for all experiments until we achieve a full connectivity of the network.

To begin, we measure the number of nodes connected NNC in the DODAG when ring size is fixed as per the values of experiments set in Section 4.2 for different networks. We then measure the number of nodes connected securely NNCS that are in the DODAG and share a key. This is essential for this research as any two nodes that do not share a key and cannot communicate securely need to be discarded. This has a bigger effect on networks that uses RPL since discarding any node will discard all its child nodes even if they can themselves communicate securely because they share a key. We observe in this experiment that the number of nodes that are in the DODAG (annotated number of connected nodes NNC) is relatively high regardless of the size of the ring or the size of the pool, however, the number of nodes that are in the DODAG and share a key securely (number of connected nodes securely NNCS) falls drastically to nearly 50% of the nodes in most networks or even less in smaller network as shown in Figure 4.3. We also observe that the number of nodes that share a key fluctuate between OF0 and ETX. When using ETX, it is expected that the hops count will increase as it is not an essential metric for this objective function since it relies on the nodes metrics such as power consumption and CPU usage in comparison with OF0 that relies on hops count. It is also observed that this does not have an impact on the number of nodes that can communicate securely and are in the DODAG.

We experiment with the ring size for both RPL ETX and OF0 objective functions, we note in Figure 4.4 and Figure 4.5 that the ring size decreases for one pool size when the network size increases. For example, for a pool that contains 100 keys and identifiers *RS* needs to be 51 to achieve full connectivity if the network size is of 10 nodes. When we increase the number of nodes in the network, the ring size decreases so for a network of 100 a ring size of 29 keys achieves full connectivity of a network of 100 nodes. We also note that that when the size of the pool becomes a lot larger than the size of the network the ring size increases exponentially. In ETX, the ring size reaches 197 keys and identifiers for a 100 nodes network when the pool is composed of 2 500 keys and identifiers. That is a significant difference to the ring size of 104 keys and identifiers when the network is of 2 500 nodes.

In Table 4.2 we compare the ring sizes obtained to achieve full connectivity when using both ETX and OF0 and when pools and networks are of the same size. We observe that the rings size decrease when using ETX in comparison with OF0 to achieve full connectivity.

We note that the ring size needed to achieve full connectivity is smaller for the OF0 in comparison with ETX. This is due to the method each objective function calculates in its path. For OF0, more constraints are present to compute the path as the objective function

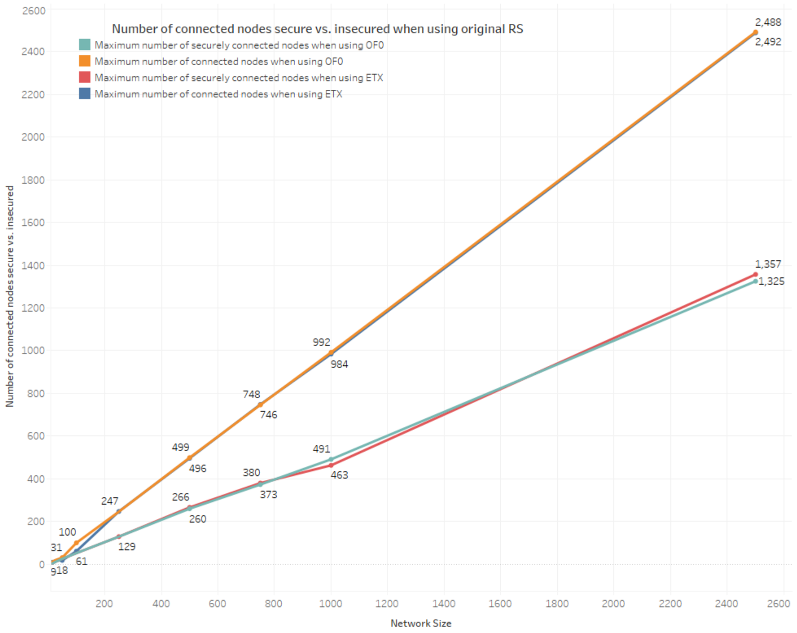


Figure 4.3: Number of insecure DAGs in the DODAG (NNC) vs the number of secure DAGs. The number of insecure DAGs in the DODAG is correlated with the percentage of shared keys negatively.

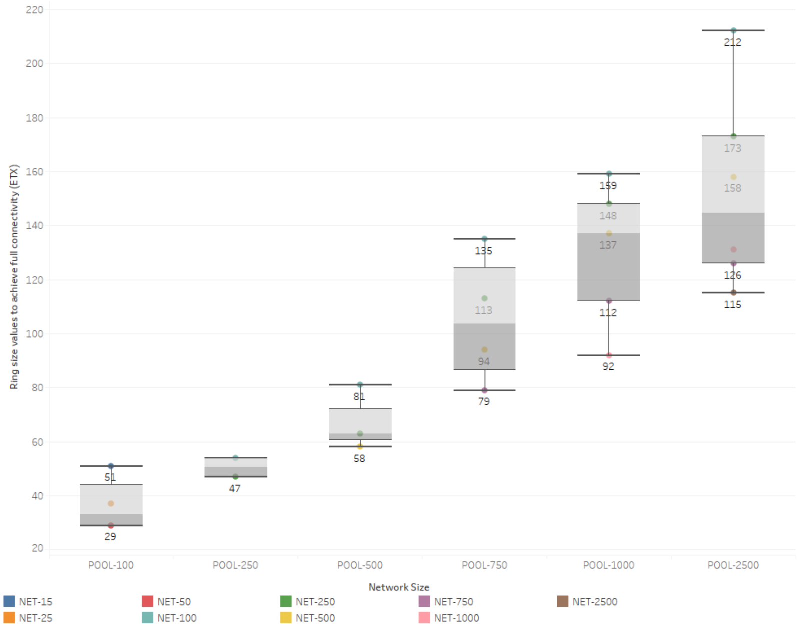


Figure 4.4: Ring size values to achieve full connectivity of network when using ETX. The ring size for each experiment was incremented for each run until each node achieved a secure link with one or more other neighbouring node in the IoT network.

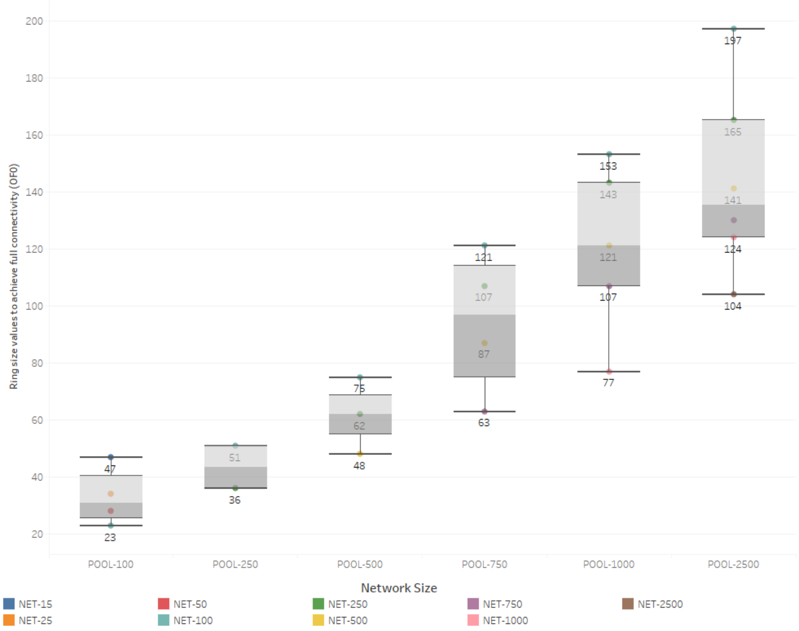


Figure 4.5: Ring size values to achieve full connectivity of network when using OF0. The ring size for each experiment was incremented for each run until each node achieved a secure link with one or more other neighbouring node in the IoT network.

Table 4.2: The Ring size values for Probabilistic scheme for various pools when the network size is the same. That is the ring size for experiments where a network and a pool are of the same size, i.e. 2500 nodes in a network and pool size of 2500 keys.

Original values				ETX	OF0
P	N	RS	SK %	RS	RS
100	100	8	50.52	23	29
250	250	13	50.43	36	47
500	500	18	57.14	48	58
750	750	22	49.47	63	79
1000	1000	25	57.14	77	92
2500	2500	41	48.19	104	115

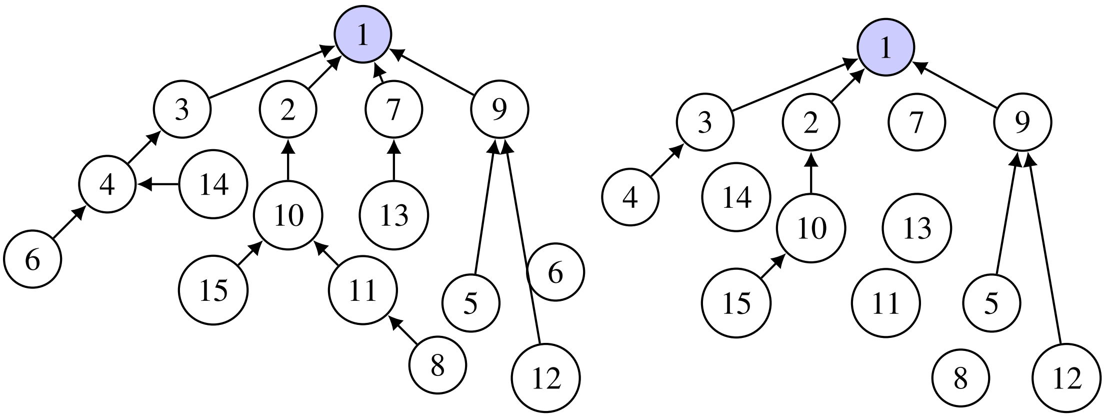
uses link metrics only and more specifically hops count. This means that nodes will only need to find one or more preferred parent with the shortest path to the root. In ETX, nodes choose their parents based on the node metric. More nodes will potentially be the preferred parent for a node when using ETX and the node can discard one even if it shares a key with it because the node metric are not desirable. This will naturally lead to larger rings size as nodes will need to achieve two constraints rather than one and share a key and have an acceptable node metric.

We have identified the ring size required to achieve a complete connectivity where all nodes in the Distributed Sensor Network (DSN) are calculated, we can identify how it will perform in an IoT environment using routing protocols designed specifically for the IoT. We performed several experiments using those values when RPL Objective Function zero (OF0) is applied and when the Expected Transmission count (ETX) is applied.

4.4 Fixed Network Experiment

To gain a clear view of why this is happening we experiment on a small network of 15 nodes in a fixed network environment where we distribute a ring size of 8 keys in all nodes following the experiment setup introduced in Section 3.7. We know that those nodes operate with a range of 50 meters. We put them in a closed environment where the density is high (50x50 meters). We simulate the network first with no keys and all nodes can communicate insecurely. The DODAG formed when using ETX is shown in Figure 4.6a and OF0 in Figure 4.7a. We then load the rings listed in the Appendix C Section 3.2 for nodes 1 to 15. We note that when using ETX objective function that only 9 nodes join the secure DODAG as shown in Figure 4.6b and 6 are discarded. We repeat the same experiment with the OF0 objective function shown in Figure 4.7b and we note that although the topology changes the number of nodes that are discarded does not change in our fixed experiment environment. We also note for both ETX and OF0 that if one node is discarded from the secured RPL all the leaves of that node are also discarded which is expected since for a node to be in the DODAG it needs to have a complete link to reach the root node.

We also note that the number of connected nodes that are secure do not reflect the number of nodes that share a key. In a normal environment, two nodes that share a key can form a DAG. In an IoT environment and because RPL is using the secure connectivity for each node it is only stored in the DODAG if the whole path is secure and not only one DAG. This means that many DAGs can potentially be discarded even if they are secure but because their parents cannot find a shared key with its ancestor means it cannot form a secure DAG to the root. This is the same as the example given in Figure 4.7b.

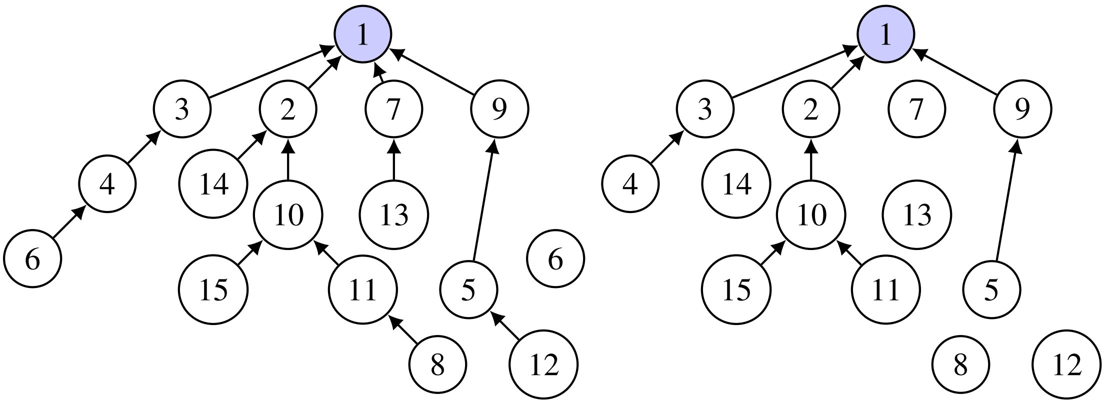


(a)
Insecure DODAG using ETX.

(b)
Secure DODAG using ETX.

DODAG using ETX, first in Figure 4.6a where the mechanism of secured communication is not applied and keys are not distributed and all nodes are participating in the Internet of Things network and second in Figure 4.6b where network does not include all nodes and all nodes that do not share a key and the DAG is discarded. Only 10 nodes out of 15 nodes only participated in the secure DODAG.

Figure 4.6:]



(a)
Insecure DODAG using OF0.

(b)
Secure DODAG using OF0.

Figure 4.7: DODAG using OF0, first in Figure 4.7a where the mechanism of secured communication is not applied and keys are not distributed and all nodes are participating in the Internet of Things network and second in Figure 4.7b where network does not include all nodes and all nodes that do not share a key and the DAG is discarded. Only 10 nodes out of 15 nodes only participated in the secure DODAG.

In the next section, we investigate the impact of increasing the size of the key ring and identifier ring that is distributed for all nodes.

4.5 Evaluation Of The Impact of Increasing RS In An IoT Environment

In this section, we evaluate the impact of increasing the ring size in the IoT environment by using the ring sizes obtained in Figures 4.5 and 4.4. We investigate the overhead of the ring sizes increase in term of the number of RPL control messages, the average hop counts, the power consumption, the CPU usage, the time to converge, the latency and the number of neighbours.

4.5.1 RPL Control Messages Number

In this section we look at the RPL control traffic overhead generated for nodes. We ignore overheads that are not related to the formation of a secure DODAG or any other data traffic. We measure the number of RPL control traffic messages DIO, DIS and DAO when using OF0 and ETX objective functions over 24 hours to maintain consistency and to ensure that RPL DODAG has converged and little changes are occurring.

In both Figure 4.8 and Figure 4.9 the largest overhead in the three control messages is the number of DIO messages. DIO messages are messages that will be generated by each node to signal its presence and to broadcast its metrics regardless of the objective function in use in order for child nodes to use it as a parent. The second in order of the number of control messages sent is the DAO messages. DAO messages are destination advertisement object messages that are sent for each node to a node that has a higher rank in order to attempt to use it as a parent node. The number of DIS packets is the lowest. DIS messages are used when a node joins a network and this is only done once a DODAG is formed. Once a node joins the DODAG, there will be no need for it to send to the DODAG root another DIS.

We observe that the number of control messages is not directly related to the pool size however since we are ensuring a secure DODAG, the number of hops is naturally increasing and DAO messages that find that it does not share a key needs to be discarded thus increasing the number of control messages and more specifically the number of DIO and DAO messages. This implies that the increase in the number of control messages is indirectly related to whether two nodes share a key or not.

We also note in both Figure 4.8 and Figure 4.9 that the number of RPL control messages under OF0 is larger than when using ETX. The calculation of the most suitable path using

OF0 needs more control messages than ETX in a dense network. This means that when the number of nodes increases in OF0 more RPL control messages are needed to select the best route. This explains and justifies the number of RPL control messages in OF0 in comparison with the OF0 RPL control messages.

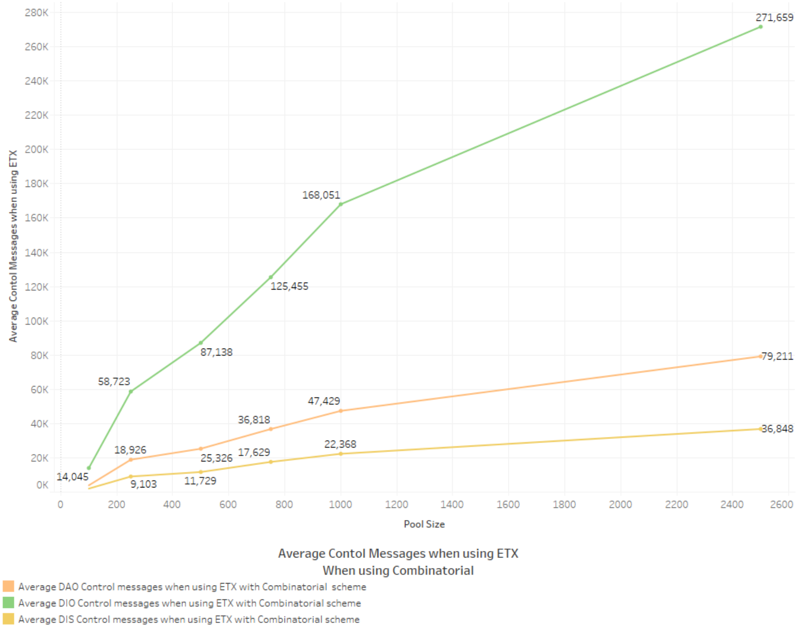


Figure 4.8: Number of all the RPL Control messages (DAO, DIO, DIS) generated in 24 hours for different networks using ETX. The number of DIO messages is the largest since DIO messages are sent for all nodes regardless whether the nodes share a key or not. The DAO nodes are only generated as a response to DIO messages if the nodes share a key and the DIS messages are only generated if the node is selected a preferred parent.

Figure 4.10 shows that for each network size there is an increase in the number of control messages between ETX and OF0. ETX outperforms OF0 in all RPL control messages as the lesser the number of RPL control messages there is the less overhead we are adding which in turn means there will be less power consumption.

Another factor that we need to look at is the number of messages that are lost due to collisions and retransmissions. This also increases the overhead specifically in dense networks where collision is a lot higher. To look at the impact of messages needed to be retransmitted due to collision and loss, we observe the packet delivery ratio for control messages. Packet delivery Ratio (*PDR*) is the ratio of number of packets that were delivered over the number of packets that were sent in total. The higher the PDR value is the lower the value of overhead loss due to collision. Ideally, all messages that were sent will be delivered and this leads to a *PDR* value of 1.

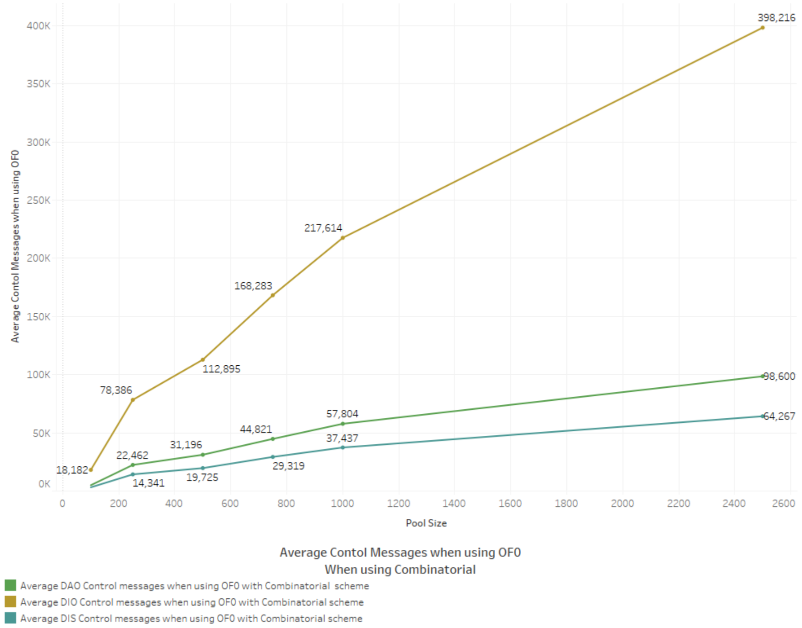


Figure 4.9: Number of all the RPL Control messages (DAO, DIO, DIS) generated in 24 hours for different networks using OF0. The number of DIO messages is the largest since DIO messages are sent for all nodes regardless whether the nodes share a key or not. The DAO nodes are only generated as a response to DIO messages if the nodes share a key and the DIS messages are only generated if the node is selected a preferred parent.

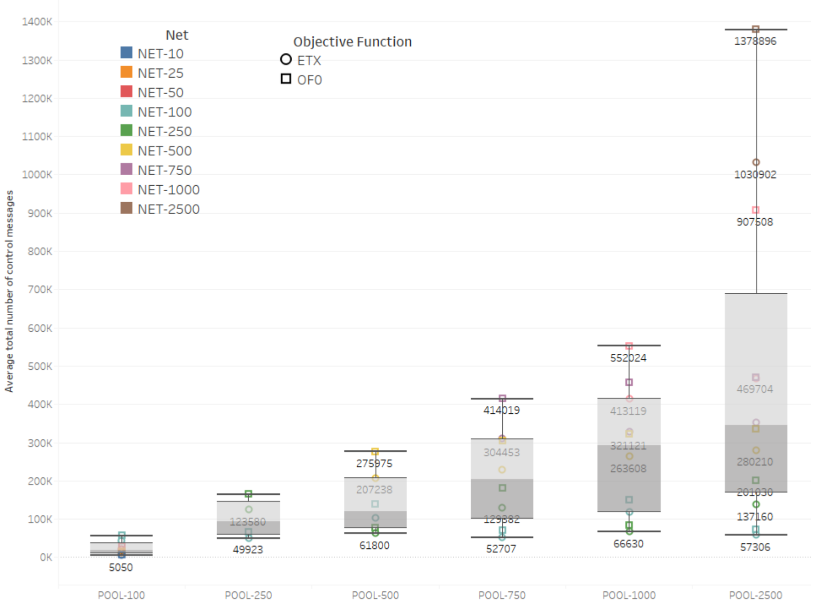


Figure 4.10: Total number of RPL Control messages generated in 24 hours for different networks and pools sizes when using ETX and OF0 (Probabilistic). For each network size there is an increase in the number of control messages between ETX and OF0

The higher the value of PDR, the lower the value of retransmission which leads to less resource waste. Figure 4.11 shows the variation of PDR value with respect to the network size using OF0 and ETX. It shows that as the network size increases, the PDR decrease and ETX outperforms OF0 in terms of the number of wasted messages due to collision. We note for example that when the number of nodes reaches 2 500 OF0 delivers only 18% of the control messages while ETX still delivers 38%, an outperformance of 20% in the PDR.

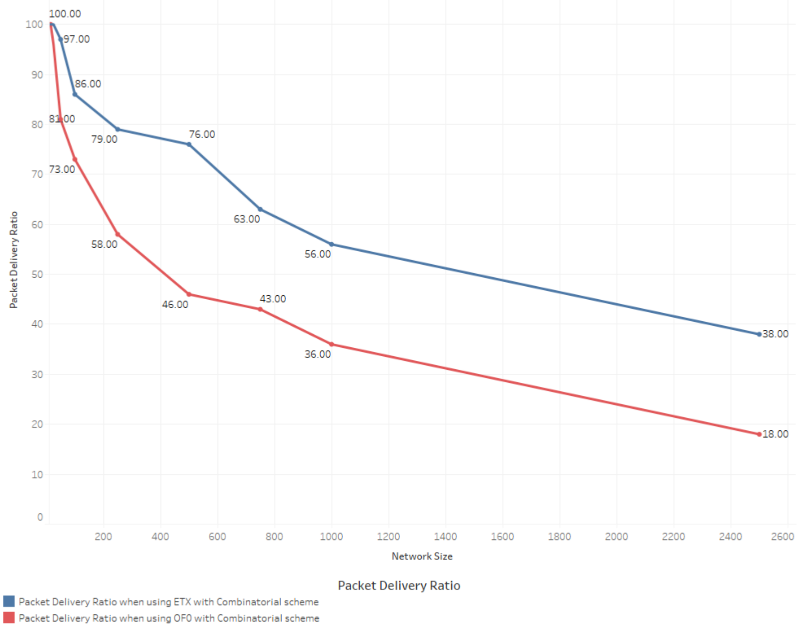


Figure 4.11: Packet Delivery Ratio of control messages: OF0 vs ETX. We observe that the ETX objective function outperforms OF0 for all networks size however the packet delivery drops when the network size grows.

4.5.2 Hops Count Average

Hops count is an important metric when evaluating the performance of one objective function in RPL. Hops count will also determine the number of DAGS that a message needs to hop before it reaches the root destination of the DODAG. Hops count is a more important factor if using objective function OF0 as when a node calculates the best path to reach the root it prioritizes hops count as the first metric for this. This is how a link metric objective function selects its best path. This is in contrast with the ETX objective function which calculates the best path using node metrics.

In this section we investigate the two variables defined in Chapter 3 for the average hops count variables, initial average hops count *IAHC* and converged average hops count *CAHC*.

We note in Figure 4.12 and figure4.13 that the average hops count increases when the number of nodes in a network increases. We also note that for each network set when the pool size increases, average hops count increase too. We also note for both ETX in Figure 4.12 and figure 4.13 OF0 that the average hops count keeps on increasing until it reaches 250 nodes. For larger networks of 500 nodes and 750 nodes the average hops count for both OF0 and ETX decreases slightly then it starts increasing again and this is due to the density of the network. When we are running the experiments on 10, 25, 50 and 100 nodes the density is relatively low and nodes are far apart in the network environment. Once the number of nodes increases, the average hops count starts to decrease since the density is higher and the nodes range overlaps allowing a node to choose between two or more nodes that it shares keys with. The average hops counts increases again for networks with larger than one thousand nodes as nodes become more demanding as the options for a parent node increases and different metrics become important in the process of calculating the best path.

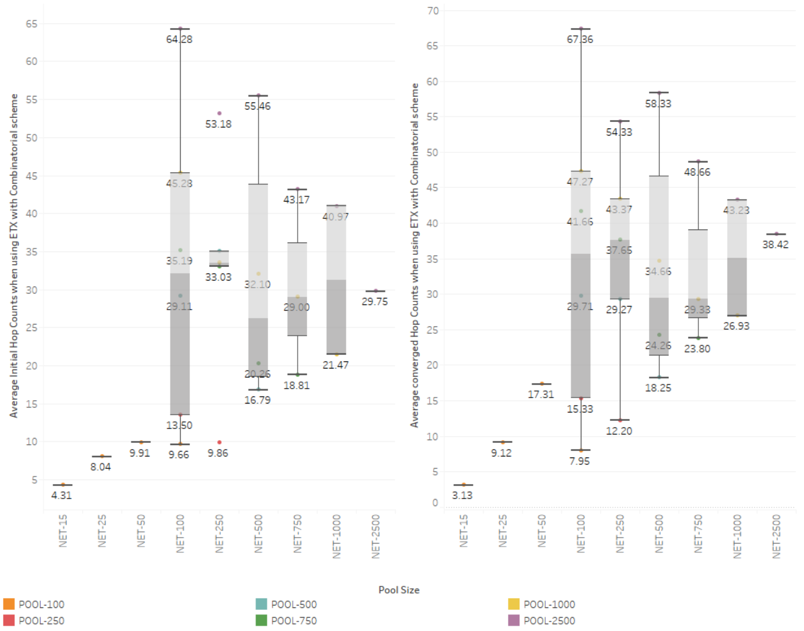


Figure 4.12: Initial and converged average hops count using RPL-ETX. The average hops count increases when the number of nodes increases in a network and the pool size increases.

We also note in Figure 4.14 the average hops count when using ETX is slightly larger than the one for OF0. This is an expected result considering the difference between ETX and OF0 as the first is a node metric objective function and the second is a link metric. A link metric objective function will prioritize the link variables such as hops count over any other node metric such as power consumption.

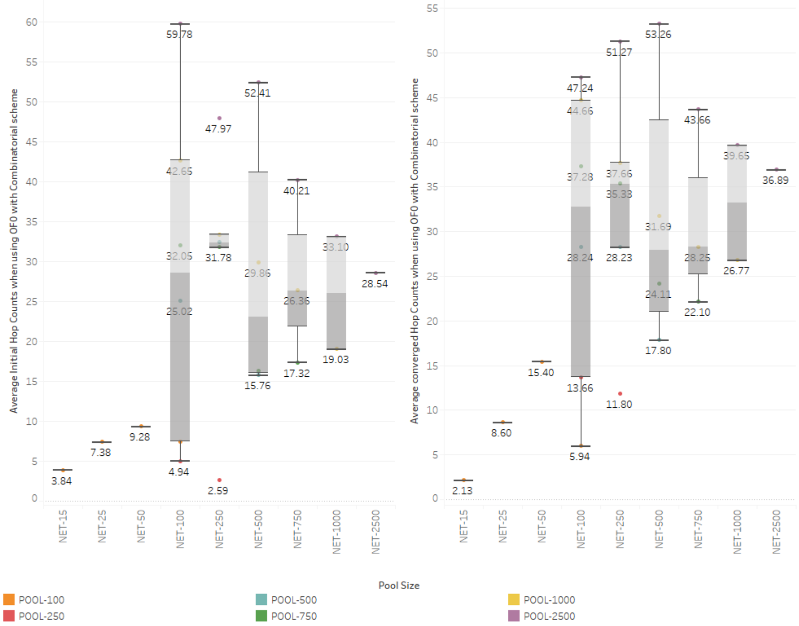


Figure 4.13: Initial and converged average hops count using RPL-OF0. The average hops count increases when the number of nodes increases in a network and the pool size increases

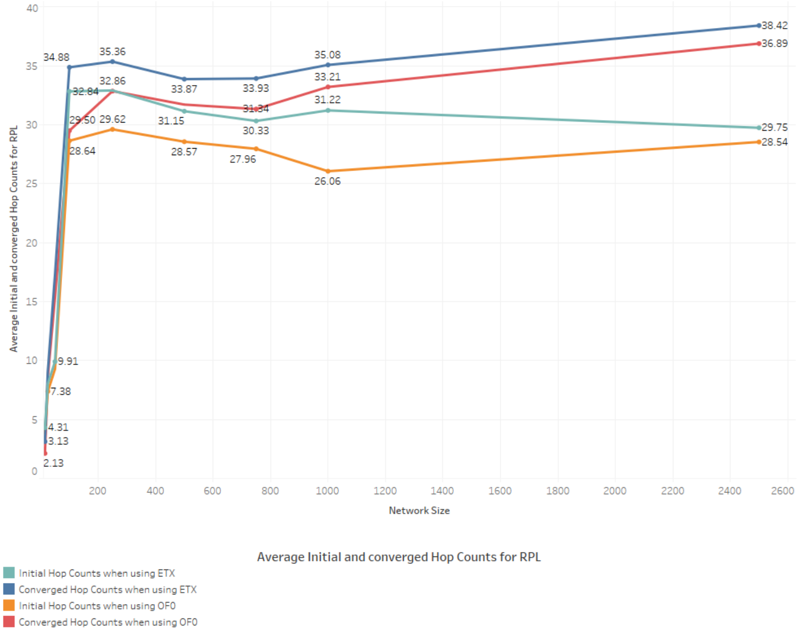


Figure 4.14: Comparison of the initial and the converged average hops count for OF0 and ETX. The ETX objective function outperforms OF0 for both the Initial and Converged hops due to the transmission count variable.

4.5.3 Power Consumption

Power consumption is considered as one of the most critical constraint of any IoT device. We evaluate in this section how the increase of the number of keys and identifiers in the ring and the overhead needed to achieve a full connectivity of a secure DAG. As explained in Chapter 3 we obtain the power consumption values for each node by extracting information from PowerTrace and importing it into the Energest tool. Information about the power consumption for each node are extracted in all experiments. We measure the power consumption when different components of each node are on. We compare the power consumption when the radio component is transmitting (Tx), receiving (Rx) and when the radio is idle or low power mode (LPM). We also measure the power consumption when the CPU is working either to generate a control message or to receive and compute a control message in order to make a decision for its position in the DAG.

Figures 4.16 and 4.15 show that the major consumption of energy is when the the radio state is either on radio transmitting (Tx) or radio receiving (Rx). It also show that the power consumption when LPM state is on and when the CPU is on are considerably low and do not increase when the number of nodes in the network increases.

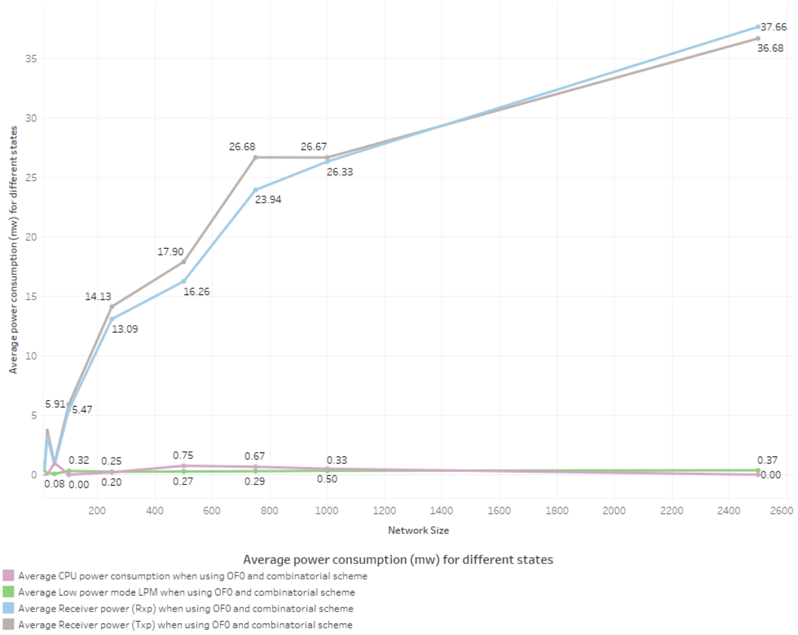


Figure 4.15: States power consumption when using OF0. The CPU and the LPM power consumption are very low in comparison with the Transmission and Receiver power consumption. The power consumption for sending and receiving increase when the network size increases.

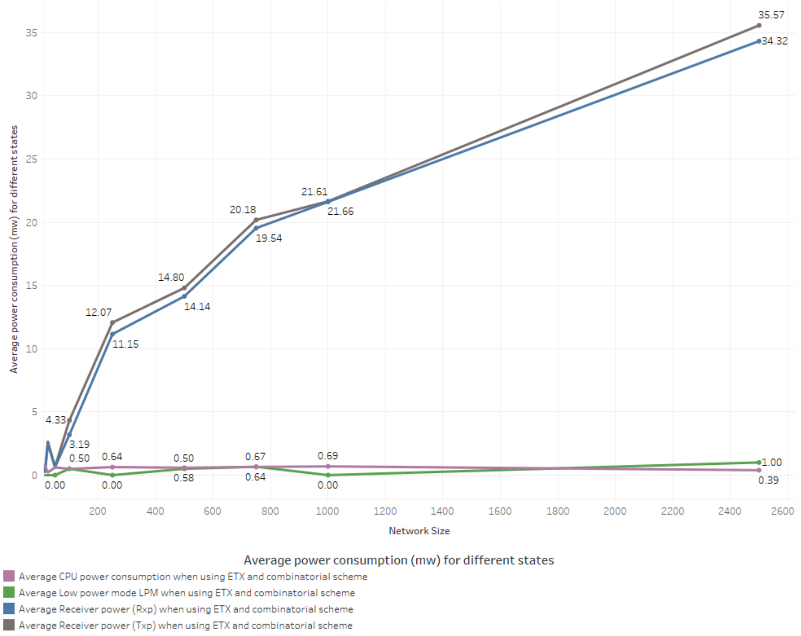


Figure 4.16: States power consumption when using ETX. The CPU and the LPM power consumption are very low in comparison with the Transmission and Receiver power consumption. The power consumption for sending and receiving increase when the network size increases.

We observe in both figures that the radio on time increases as the network density increases. When the number of nodes in the network increases, nodes become closer to each other (denser network) and more control messages will be needed for a complete secure DAG to form. The number of of control messages in a network increases as the number of nodes increases as we have explained in Section 4.10 and this will naturally consume more energy as the radio will need to be on to transmit those messages or to receive them.

In Figures 4.18 and 4.17 we evaluate the power consumption for network sizes in comparison with the pool size. We observe that for a fixed network size the power consumption is higher. This is due to the fact that for a larger pool size a larger ring is used. This means that in higher density networks, nodes have more than one node that can be potentially used as a parent node. This is clear for OF0 in this figure as more nodes can be hops and each node needs to calculate the best path and most suitable parent.

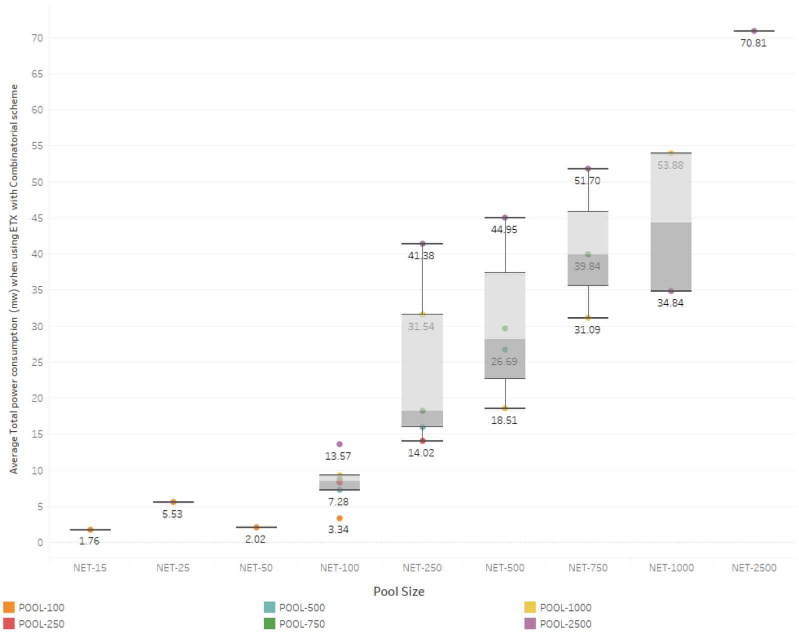


Figure 4.17: Average power consumption for ETX. Power consumption increases when the network size increase for different pool size. This is due to the fact that in larger networks, density is higher and nodes will generate larger number of RPL messages for RPL DODAG.

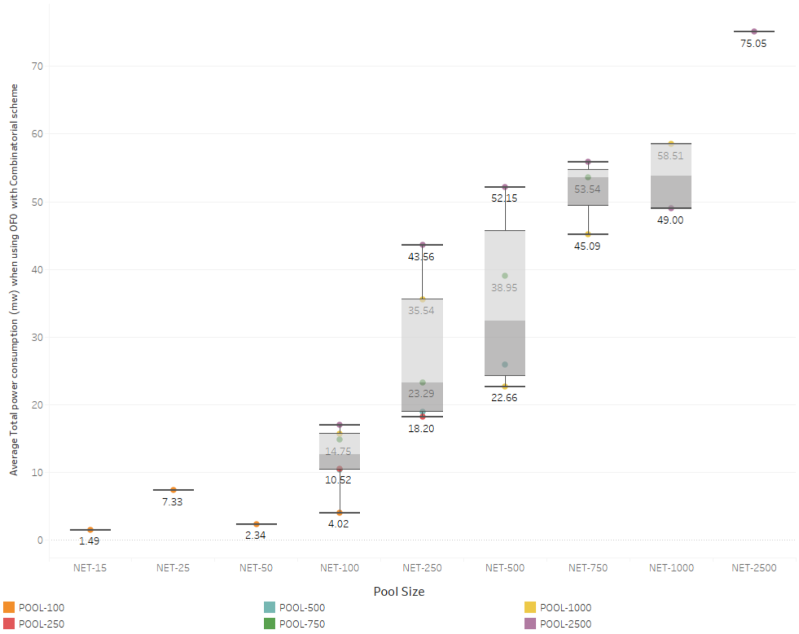


Figure 4.18: Average power consumption for OF0. Power consumption increases when the network size increase for different pool size. This is due to the fact that in larger networks, density is higher and nodes will generate larger number of RPL messages for RPL DODAG.

We also notice in Figure 4.19 that the average combined power consumption for all nodes (APC) when using the ETX objective function than when using OF0. This is due to the fact that when ETX selects best path it chooses the power usage for each node as a metric to select the most suitable parent in comparison with the objective function OF0 that chooses the best path by looking at the hops count and rank of a node before it considers it a parent.

We compute the average duty cycle as discussed in Chapter 3 Equation 3.5.16 where the average duty cycle is the sum of power consumption of both transmitter and receiver radios divided by the power consumption of the CPU and and the LPM states. The transmitter Tx and the receiver Rx are the two components that consume the most as we have identified before. Therefore, the average duty cycle will be larger for the objective function that consumes more power. We compare the average radio duty cycle *ardc* in figure 4.19 for our experiments and we note that the average radio duty cycle increases when the density of the network increases. This result validates our experiments in terms of the difference in power consumption between ETX and OF0.

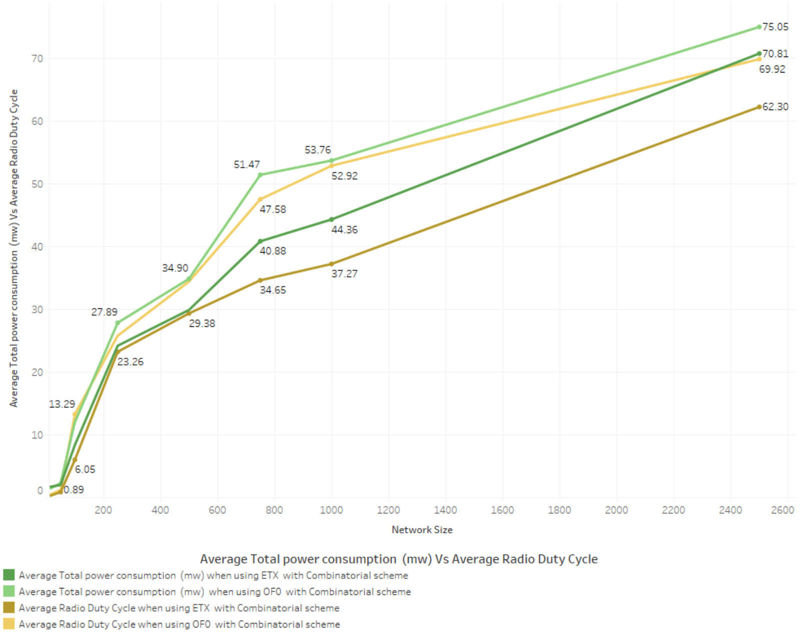


Figure 4.19: Average power consumption vs average radio duty cycle for OF0 and ETX. ETX objective function outperforms OF0 for the power average consumption. This is due the that fact that the ETX objective function uses the transmission Count TC variable to find a node to the root of the DAG instance with the least number of transmissions. The average Radio Duty Cycle RDC for ETX outperforms the OF0 objective function since RDC when using ETX will switch the radio off more because of the transmission Count TC .

4.5.4 CPU Usage

In this section we measure the maximum CPU usage a node reaches both when forming the RPL DODAG and after the RPL converge. We then compute the initial average CPU usage and the converged average CPU usage for all nodes. We note that the initial CPU usage is bigger than the converged CPU usage when using both ETX and OF0. We also note in Figure 4.20 and Figure 4.21 that for each network size the initial CPU usage and the convergence CPU usage do not vary a lot. For example for a network that has 100 nodes the average initial CPU is around 30% and is not affected by the size of the pool.

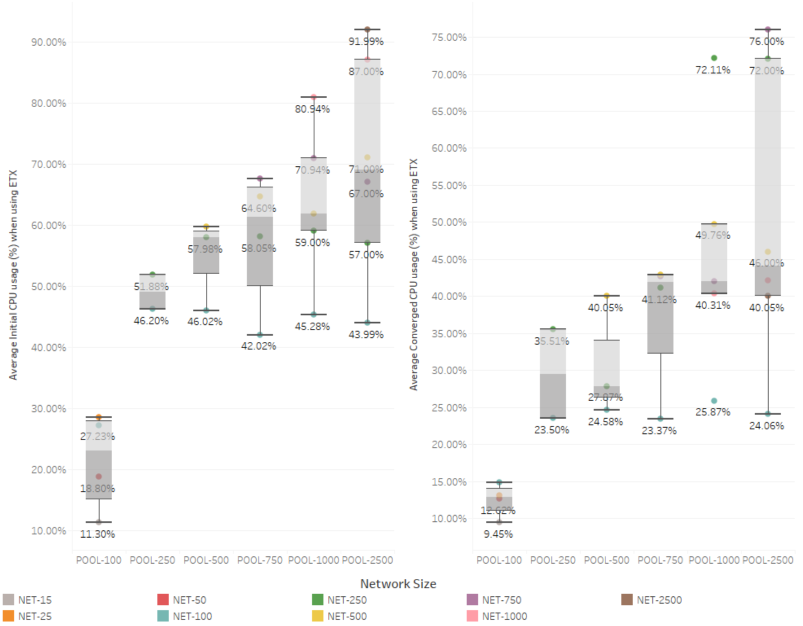


Figure 4.20: Initial CPU usage and converged CPU usage for RPL using ETX for different pools size. The CPU usage increases when the network size increases regardless of the pool size. The CPU usage decreases when the DODAG converges.

We compare in Figure 4.22 the average initial CPU usage and the average converged CPU usage for both ETX and OF0. We can see that the objective function OF0 outperforms ETX in of the CPU usage both for the initial and the converged CPU usage. We note that the average CPU usage percentage when using ETX is larger than when using OF0 both during the initial DAG formation and after it converges. This is directly related to the nature of OF0 and the number of control messages it generates in comparison with ETX since it uses the rank to compute the preferred parent and it does not compare other link metrics as in ETX.

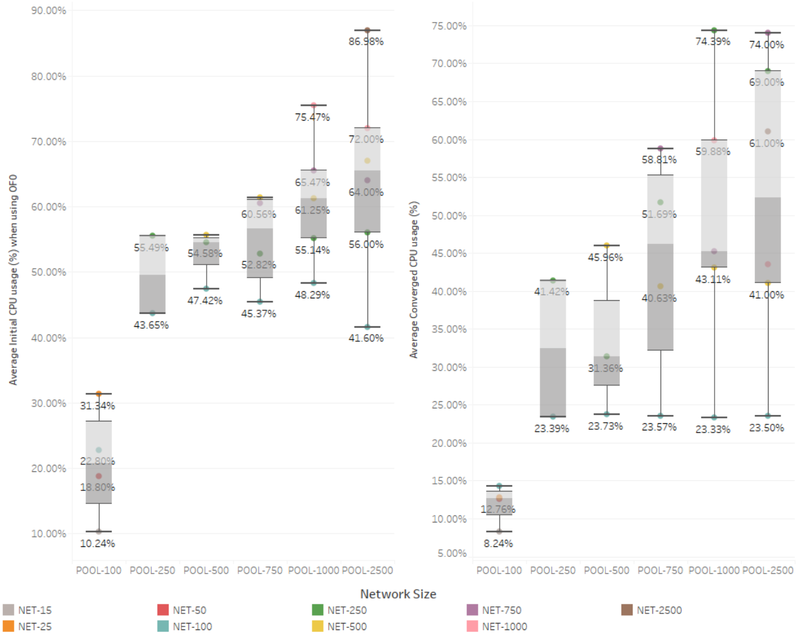


Figure 4.21: Initial CPU usage and converged CPU usage for RPL using OF0 for different pools size. The CPU usage increases when the network size increases regardless of the pool size. The CPU usage decreases when the DODAG converges.

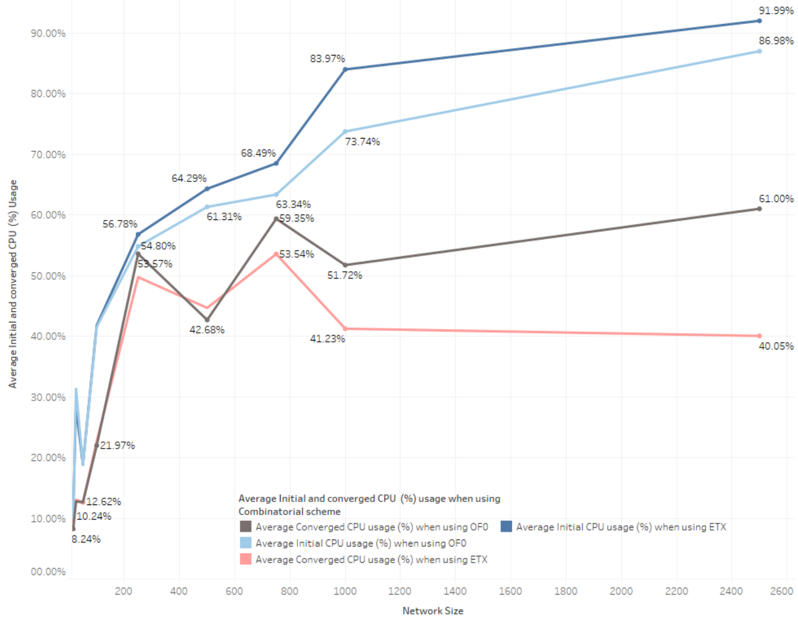


Figure 4.22: Initial CPU usage and converged CPU usage for RPL using ETX vs OF0. OF0 objective function outperforms ETX in of the CPU usage both for the initial and the converged CPU usage.

4.5.5 Time To Converge

In this section, we compare the time it takes for the DODAG to converge. We define convergence of the RPL when no significant changes happen to the DODAG. To ensure we achieve this we run each experiment for 24 hours. We note that the changes in the Time To Converge (TTC) is not related to the size of the pool as we can see in Figure 4.23 and Figure 4.24. We also note that the time to converge increases when the network size increases as shown in Figure 4.25 since ETX considers the transmission count variable on top of the preferred parent using the ETX path metric while OF0 only considers the neighbours with good enough connectivity. .

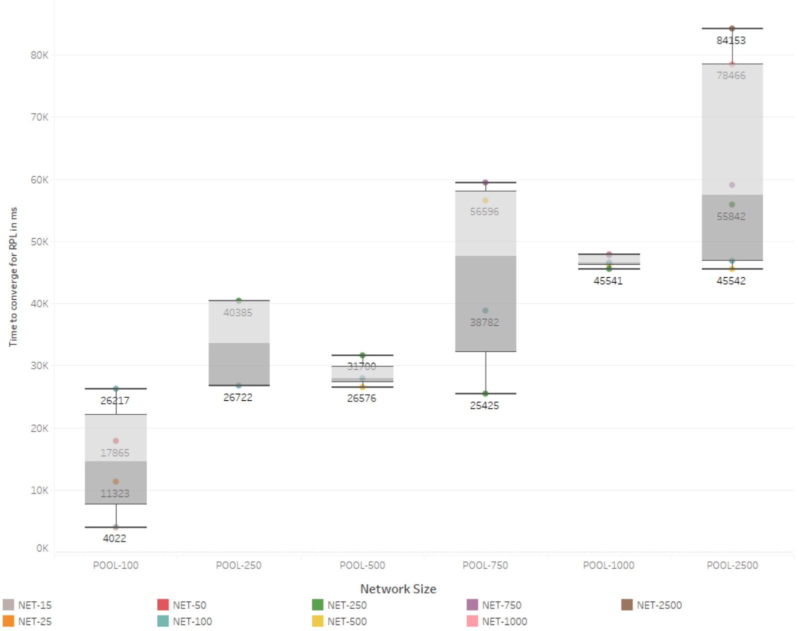


Figure 4.23: RPL DODAG time to converge when using ETX for different pools size. The time to converge increases when the network size increases.

4.5.6 Latency

We evaluate in Figure 4.26 the average network latency for packets to travel from a node and reach the root node using OF0 and ETX in the experiments environment testbed. We note that the delay of the successful packet delivery is related to the hops count and how busy the network is. We observe that the latency comparison when using OF0 and ETX is consistent with the average hop counts shown in Figure 4.14 with a considerable increase in the hops count between OF0 and ETX, the latency is still relatively the same for both OF0 and ETX. For a 2 500 nodes network the difference in the delay is not more than 0.2 ms. For

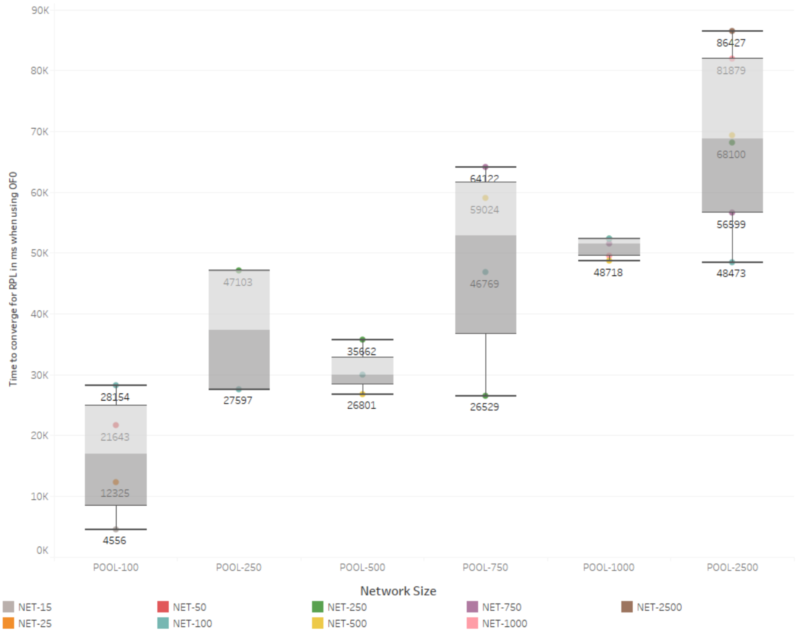


Figure 4.24: RPL DODAG time to converge when using OF0 for different pools size. The time to converge increases when the network size increases.

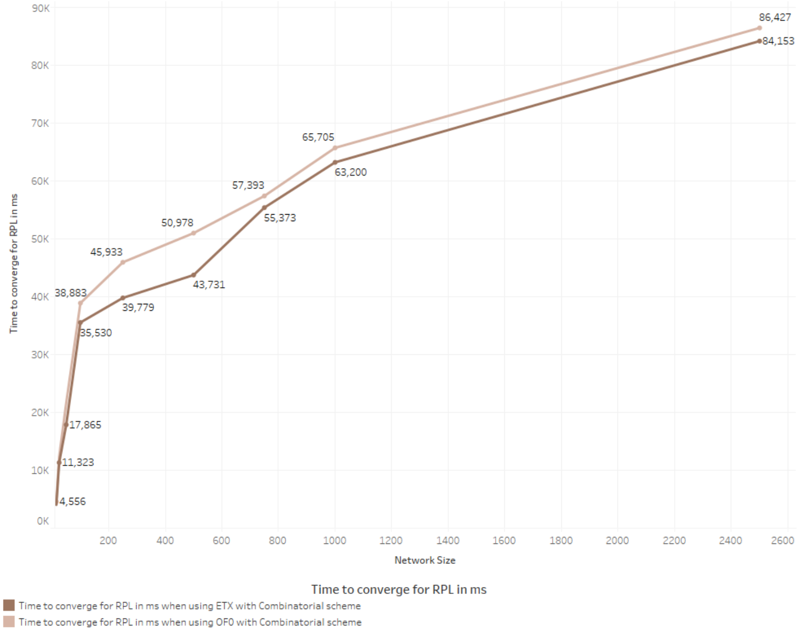


Figure 4.25: RPL DODAG average time to converge for both ETX and OF0 in different networks. OF0 outperforms ETX as ETX considers the transmission count on top of the preferred parent metric when using OF0.

smaller networks the latency is similar and the difference can only be seen in networks larger than 100 nodes.

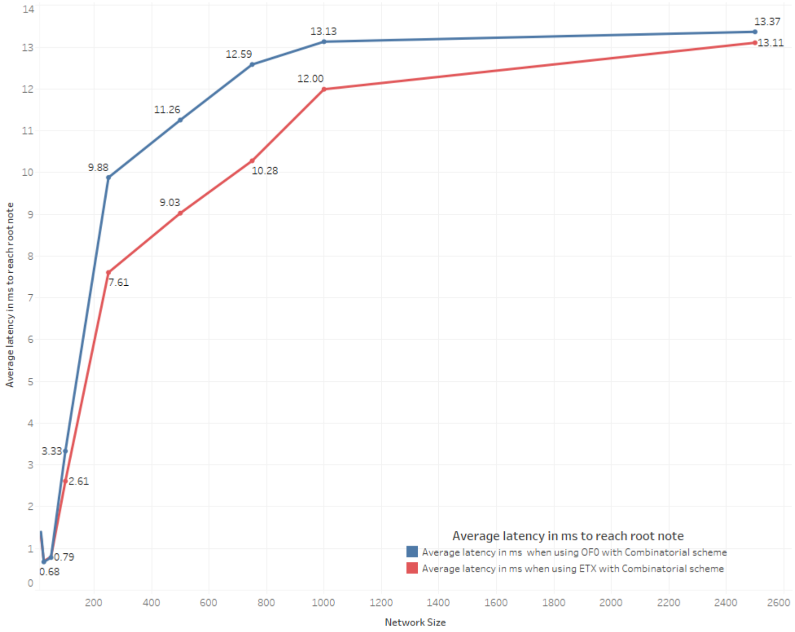


Figure 4.26: Average latency in ms when using OF0 and ETX. The latency increases when the network size for both ETX and OF0 however ETX outperforms OF0 since the transmission count TC variable in ETX discard number of nodes and allows the transmission to go through only selected nodes.

Although OF0 determine the preferred parent based on the smaller value for each neighbour the latency is slightly higher than the ETX. This is due to the fact that OF0 determine the preferred parent without taking into consideration the link quality. ETX computes the ranking by computing the path that requires the least number of delivered packets between a node and the DAG root. This is related to the Packets delivery ratio where the number of packets discarded by the number is higher in OF0 and thus the number of delivered packets ratio is smaller as shown in Figure 4.11.

4.5.7 Number Of Neighbours

The average number of neighbours as defined in Section 3.5.9 is related to the density of the network in the environment test-bed as the nodes are within range, they also need to share a key and be in the routing table. The smaller the network the more dispersed the nodes are in the 250 x 250 meters environment we have chosen to conduct all experiments. It is clear in Figure 4.27 that there is no significant difference between the number of neighbours for both OF0 and ETX.

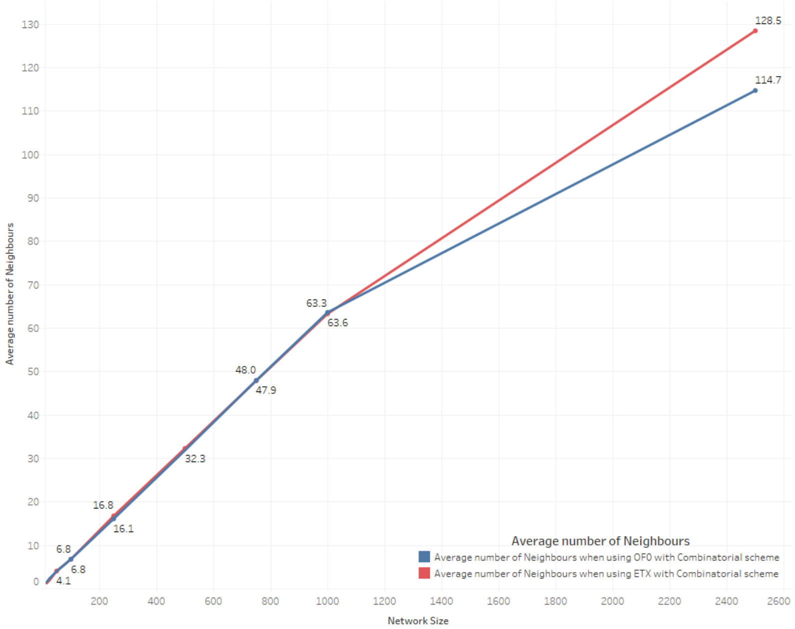


Figure 4.27: Average number of Neighbours when using OF0 and ETX. When network size increases the average number of neighbours for each node increases and the ETX and OF0 objective functions result in similar number of neighbours for smaller networks but OF0 outperforms ETX when the density of the nodes in the network decreases.

We also note that the only noticed difference in the number of neighbours is when the experiment is running in the large network of 2 500 nodes. This is due to the fact that the OF0 discards nodes with lower ranks in comparison with ETX that computes the preferred parent using the link metric and the number of delivered nodes but do discard other nodes and keep them as candidate nodes.

4.6 Summary

In this Chapter the performance of the Probabilistic key pre-distribution scheme proposed by Eschenauer and Gligor was evaluated in the context of the IoT by simulating it in the testbed designed in Chapter 3. This was achieved by first identifying how thr IoT routing protocol RPL will perform when using both ETX and OF0 while using the ring size computed in Section 4.2. We have determined in this experiment that the rings size computed from Equation 3.5.1 does not achieve full connectivity and in fact achieves no more than 54% and 53% when using ETX and OF0. The impact of increasing the rings size until the network was fully secured was evaluated in term of the overall network performance, the link metrics of all DAGs and all node metrics.

Chapter 5

Linear Key Pre Distribution Scheme

In this chapter we evaluate the performance of the Deterministic linear key pre-distribution proposed by [47] and explained in Section 2.5.5 in the context of the IoT. In the first set of experiments in Section 5.1 we investigated how the linear key pre-distribution scheme performs in term of the percentage of shared keys in the network and the number of nodes that join the DODAG by using the ring sizes calculated using the Equation 3.5.1 proposed in [46] for both ETX and OF0. We then moved to the second set of experiments in Section 5.3 where we look at how the linear key scheme performs if the ring sizes obtained in Section 4.3 and more importantly whether it achieves full connectivity of the network as well. Based on this investigation we can then carry on with the last set of experiments in Section 5.3 where we determine whether we need to increase or decrease the ring sizes and by how much. We finally evaluate the impact of using the ring sizes that were needed to achieve full connectivity of the network on both link metrics and node metrics of the nodes of the network and the links.

For the purpose of practicality, an assumption is made in terms of fingerprinting and the identification of the identity of each node. All the experiments in this Chapter assume that each identity claimed is true based on the fingerprinting provided by the analogue signal characters it presented. The aim is to achieve a 100% connectivity of all nodes in the network. Th experiment runs five times and the average is then taken for each experiment.

5.1 Network Connectivity With DSN RS In An IoT Environment

In this section we investigate the number of nodes that communicate securely using various ring sizes. We start by generating in experiment key rings and identifier rings using the *RS* variable computed in the DSN Equation 3.5.1 proposed in [46]. We have identified that the ring sizes needed for the various pool identified in Chapter 3 and shown below in Table 5.1.

Pool	Network	Original RS	PSK%	
			OF0	ETX
100	100	8	67.89	63.89
250	250	13	43.17	36.23
500	500	15	33.41	27.34
750	750	22	31.64	29.55
1000	1000	25	24.38	17.94
2500	2500	41	11.23	13.81

Table 5.1: Ring sizes for various pools using DSN calculation. ETX and OF0 do not achieve full connectivity when using those DSN for all pool and network sizes.

We observe in Figure 5.1 and Figure 5.5 that the percentage of shared keys for various pools sizes and different network sizes is very low when using ring size values computed in [46]. It is also noted that the percentage of shared keys between nodes in the RPL DODAG is higher for the small pool size of 100 keys. This applies for both ETX and OF0 where the percentage of shared keys when the density increases in the network while using 100 keys pools.

The relatively low percentage of shared keys shown in Figures 5.1 and 5.2 result in a low number of nodes joining the network and an even lower number of nodes sharing a key as shown in Figure 5.3 below.

In the next section we use the ring sizes computed from the Equation 3.5.1 and obtained in Chapter 4 with the linear scheme as shown in Figure 5.2 and Figure 5.4.

5.2 Network Connectivity With RS Values For Probabilistic Scheme

In this section, we use the same network sizes and pool sizes to experiment with the secure connectivity of the nodes in the networks when the using ring sizes *RS* values obtained

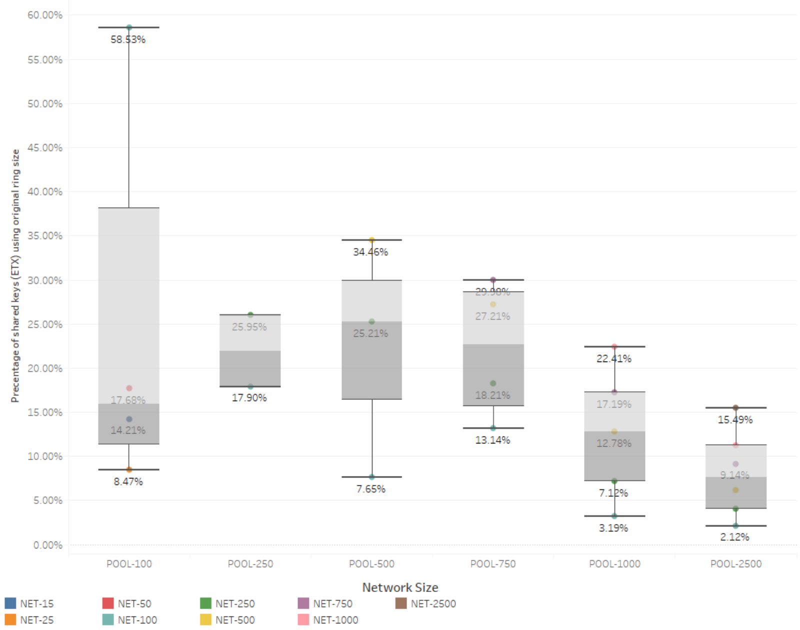


Figure 5.1: Percentage of shared keys (PSK) for different networks size and pools size when using ETX with DSN RS values. The percentage of shared keys decreases when the pool size increases and the network sizes increases as the density increases.

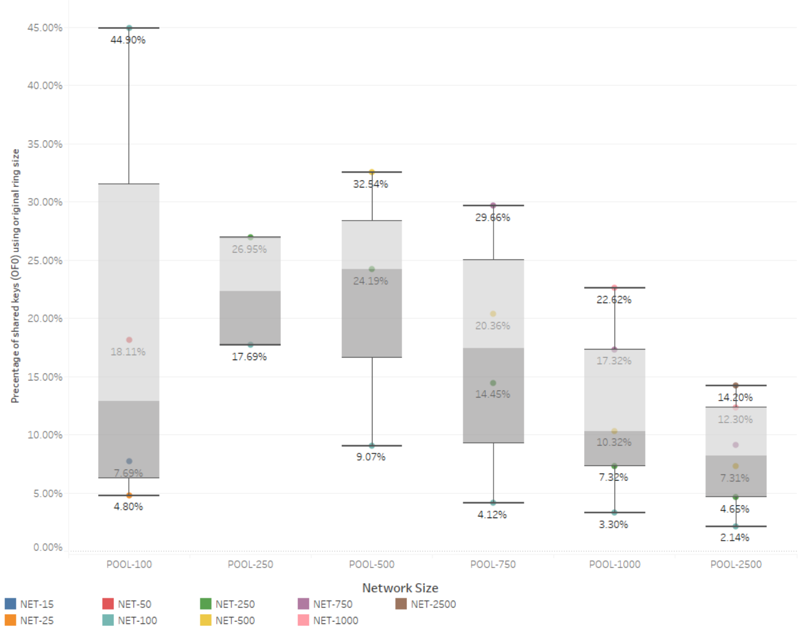


Figure 5.2: Percentage of shared keys (PSK) for different networks size and pools size when using OF0 with DSN RS value. The percentage of shared keys decreases when the pool size increases and the network sizes increases as the density increases.

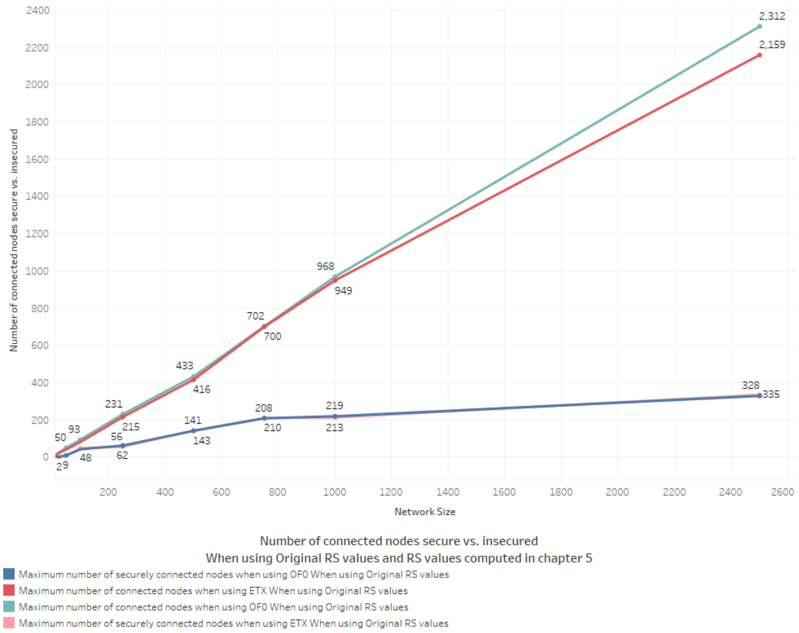


Figure 5.3: Number of DAGs in the DODAG (NNC) vs the number of secured. OF0 outperforms ETX for the number of connected nodes but the performance is nearly the same for both ETX and OF0 for the number of secured connected nodes.

in Chapter 4 to obtain full connectivity of the network. The shared key percentage obtained for those *RS* values in the context of the Deterministic Linear scheme are shown in Table 5.2 below.

When using ring sizes obtained from the simulation of Probabilistic key scheme proposed in Chapter 4 we determined that the values significantly increase but do not achieve full connectivity of the network except in small network sizes. The percentage of shared keys for different networks and different pools are shown in Figures 5.1 and 5.2. We also observe that the shared keys in the nodes in the DODAG is relatively high in OF0 and ETX for 2 500 nodes. We also observe for both ETX and OF0 that the the percentage of shared keys for networks of 500 nodes and 750 nodes outperform a larger size of networks. This is due to the fact that once networks density becomes too high such as in 1 000 nodes and 2 500 nodes the two objective functions ETX and OF0 differences will have a bigger impact on the nodes choices for parents and the computation of the path. It is also noted that Of0 outperforms ETX for all networks sizes since the selection of the path is related specifically to link metric and does not take into consideration node metrics. We can also see that the full connectivity is obtained when the pool size is of 100 keys.

Pool	Network	RS values obtained in Chapter 4 to achieve 100% connectivity			
		OF0		ETX	
		RS	PSK%	RS	
100	100	23	100	29	100
250	250	36	99.43	47	99.21
500	500	48	97.48	58	95.89
750	750	63	81.80	79	95.76
1000	1000	77	83.62	92	94.66
2500	2500	104	62.78	115	88.04

Table 5.2: Percentage of shared keys when using rings size obtained when using Probabilistic scheme in Deterministic scheme for various pools when the network size is the same. Full connectivity is only achieved when the pool size and the network size are 100.

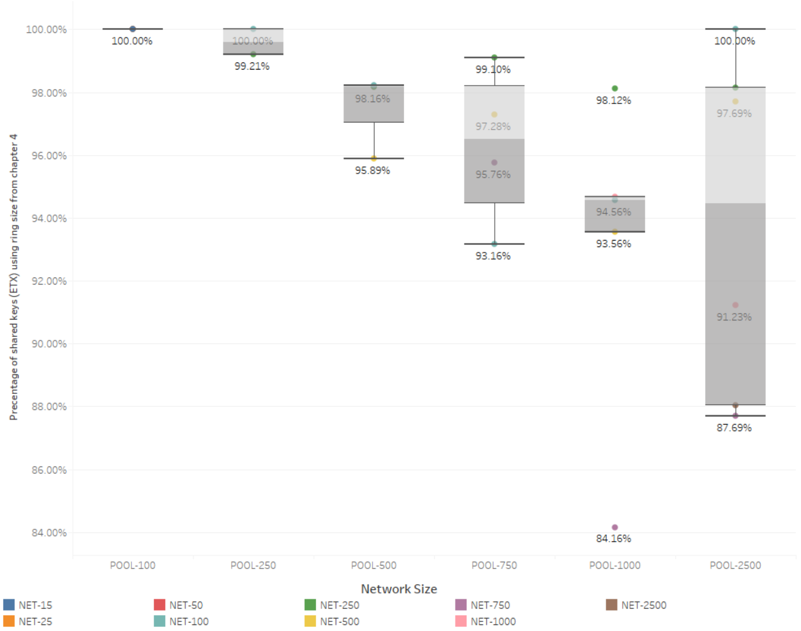


Figure 5.4: Percentage of shared keys (PSK) for different networks size and pools size when using ETX with Probabilistic RS values. Full connectivity is achieved for small pool sizes and decreases when the pool size increases and the network size increases.

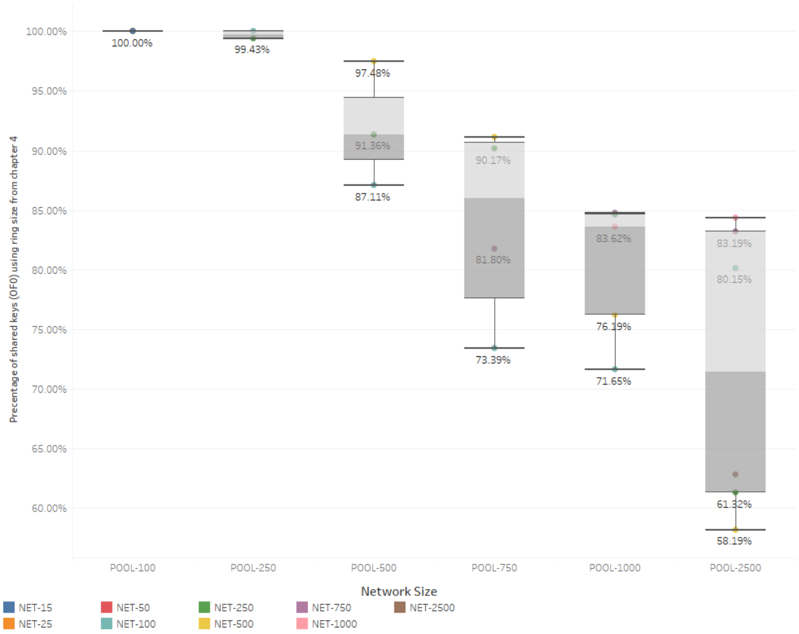


Figure 5.5: Percentage of shared keys ($PS\,K$) for different networks size and pools size when using OF0 with Probabilistic RS values. Full connectivity is achieved for small pool sizes and decreases when the pool size increases and the network size increases.

Looking at the results obtained in Section 5.1, we measured the number of DAGS available for the maximum values of ring sizes for all networks and pool sizes. We also compared the average number of DAGS in the DODAG with the number of DAGS that share a key (secure DAGS). We observed in Figure 5.3 the number of connected nodes in the DODAG and as expected it is quite low as the percentage of shared keys was quite low. This applies for both ETX and OF0. However, when the ring sizes obtained in Chapter 4 were used in the context of the linear scheme in experiment we observed that the number of DAGS increases exponentially as the number of nodes that share a key increases. The number of securely connected nodes also increased significantly but did not achieve full connectivity of the network when the pool size is of equal value to the network size as shown in Figure 5.6.

5.3 Network Connectivity With IoT RSs

Now we have identified that for larger networks using the same RS values obtained from the previous Chapter, we only achieve full connectivity in smaller networks and to achieve full connectivity for larger networks an increase in the ring sizes is needed. Similarly to what we did in Chapter 4, we keep on increasing the ring sizes in this experiment until we obtain a full DODAG that has all nodes participating in a secure way. The Table 5.3 below shows the

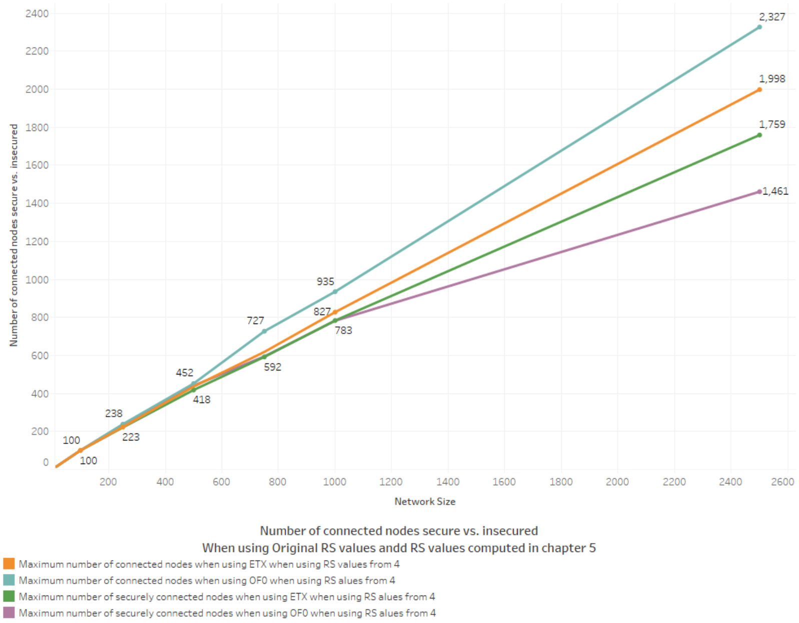


Figure 5.6: Number of DAGs in the DODAG (NNC) vs the number of secured. The number of secured DAGs in the network when using ETX and OF0 are similar for all network sizes except the large network size of 2 500 nodes.

ring size needed to achieve full connectivity of the network when using the Deterministic Linear scheme with both ETX and OF0 objective functions. The table only shows the ring size values when the pool and the network sizes are the same. The ring size values for all the network sizes and when using different pool sizes for ETX is shown in Figure 5.7 and for OF0 in Figure 5.8.

Pool	Network	ETX	OF0
100	100	40	34
250	250	38	38
500	500	63	60
750	750	86	73
1000	1000	116	114
2500	2500	136	127

Table 5.3: Rings size for Deterministic scheme for various pools when network size is the same.

We observe in Figures 5.7 and 5.8 that when comparing the ring size for different network sizes but for the same pool size the ring size value decreases when the number of nodes increases. In fact it falls by nearly half when starting with the smallest number of nodes simulated in the experiment to the equal value of the pool. For example when we start with a pool of 2 500 and 100 nodes we can see that the ring size in both ETX and OF0 is almost double than what it is when we reach 2 500 nodes in the network for the same node. We also note that the ring size for all network sizes when using ETX is larger than when using OF0 except when the pool size is smaller and the network is small. This is due to the fact that ETX uses the link metric which adds another parameter to the calculation of the preferred parent in comparison with OF0 that uses the most feasible parent without optimizing the link. The ring size values in smaller networks is larger in OF0 since the link metric in smaller networks has a smaller impact in comparison with a larger network where OF0 is most suitable.

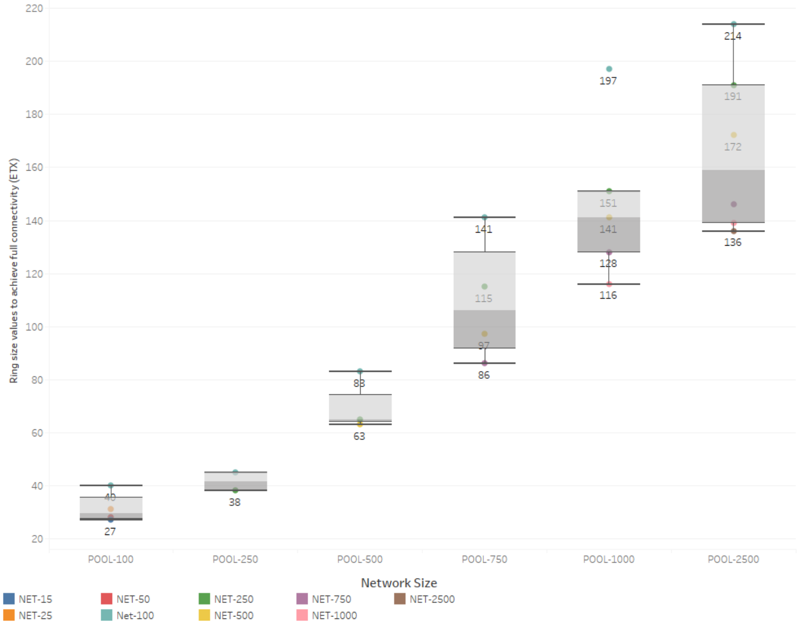


Figure 5.7: Ring size values to achieve full connectivity of network when using ETX. The ring size was increased gradually until full connectivity of the network was achieved.

We observe in Figure 5.9 that the ring size increase for ETX is larger than the increase for OF0. This is similar to the results obtained in Chapter 4. OF0 perform better in smaller network environments since the preferred parent is chosen based on the rank and is not based on the link metric.

Another essential factor that has a high impact on the increase of the ring sizes is that the Linear key distribution nodes also consider the voting process to identify honest nodes. This interferes directly with the nodes that might not achieve this requirement and thus are not selected as preferred parent. When nodes are identifying preferred parent in order to compute the best path to reach root nodes regardless of which objective function, each of them go

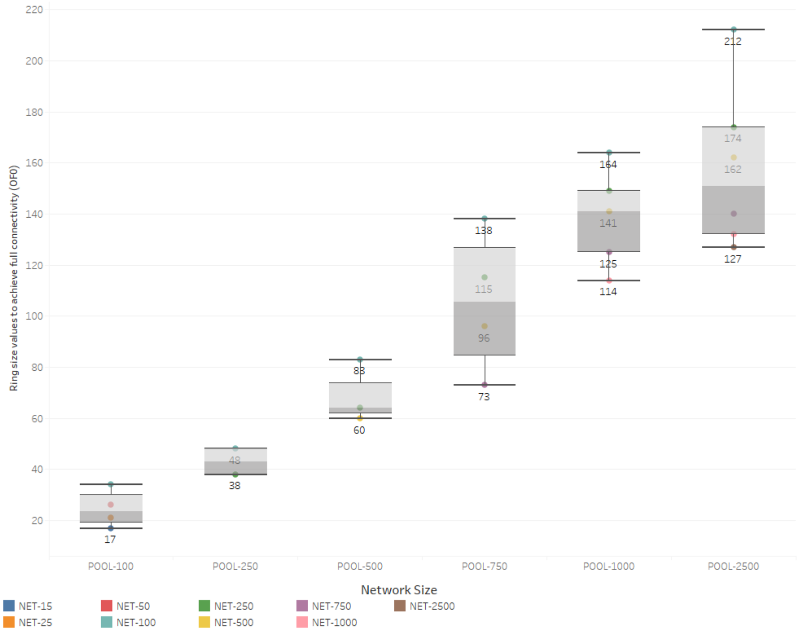


Figure 5.8: Ring size values to achieve full connectivity of network when using OF0. The ring size was increased gradually until full connectivity of the network was achieved.

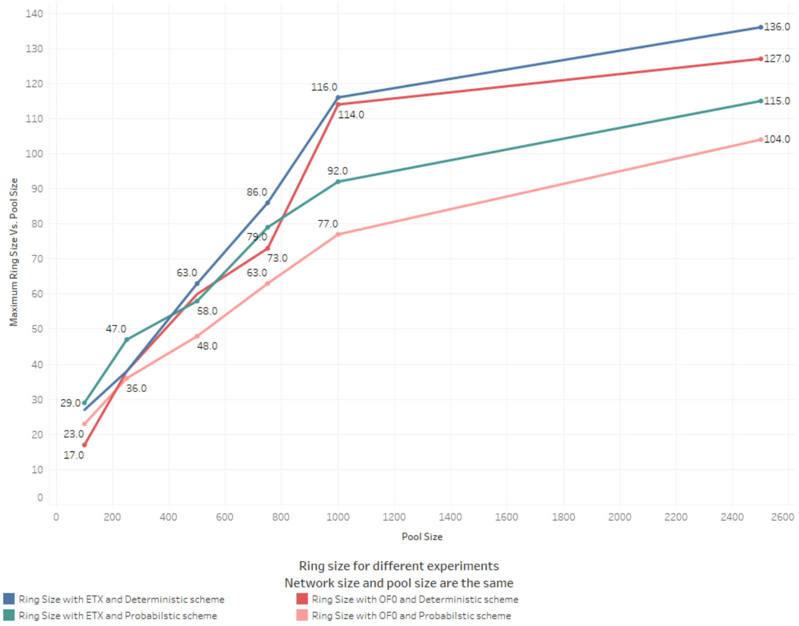


Figure 5.9: Ring size values to achieve full connectivity of network for ETX and OF0 computed in both Probabilistic and Deterministic schemes in Chapters 4 and 5. The ring size for the Probabilistic scheme outperforms the Deterministic scheme for both ETX and OF0. ETX outperforms OF0 for both schemes.

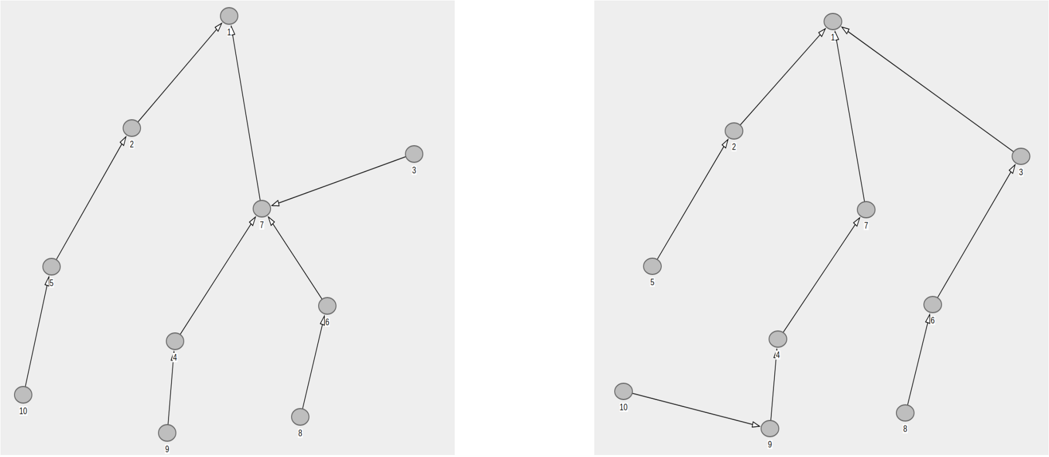
through a voting process with all of its neighbours to vote which node is mostly trusted and honest. This by itself discards some nodes to be chosen as preferred parents for others and thus one node will need to find an alternative. This will naturally cause an increase in all ring size values.

To have a better picture of what is happening when using the RS values obtained in Chapter 5 we observe the same DODAG simulated in Figure 5.10. We can see that the DODAG is fully connected however the DAGS relation between different nodes changes before we provide the different rings in the nodes and after. In Section 5.5 we evaluate the average number of hops to reach the root node as shown in fig 5.19. This is reflected in the smaller network simulation in Figure 5.10a where the number of hops increased in comparison with the Probabilistic simulation of the same network as shown in Figure 5.10b.

Looking at the results obtained from the experiment in Section 5.1 we measure the number of DAGs available for the maximum values of ring sizes for all networks and pools sizes. We compare the average number of DAGS in the DODAG with the number of DAGS that share a key (secure DAGS). We observe in Figure 5.3 the number of connected nodes in the DODAG and as expected it is quite low as the percentage of shared keys was quite low. This applies for both ETX and OF0. However when the ring sizes obtained in Chapter 4 were used in the context of the linear scheme in the experiment in Section 5.2 we observed that the number of DAGS increases exponentially as the number of nodes that share a key increases. The number of securely connected nodes also increased significantly but did not achieve full connectivity of the network as shown in Figure 5.6.

Now we have identified that for larger networks using the same RS values obtained from the previous Chapter, we only achieve full connectivity in smaller networks and to achieve full connectivity for larger networks an increase in the ring sizes is needed. Similarly to what we did in Chapter 4, we keep on increasing the ring sizes in this experiment 5.3 until we obtain a full DODAG that has all nodes participating in a secure way. We note in figure 5.9 that the ring sizes increase for ETX is larger than the increase for OF0. This is due to the fact that OF0 outperformed ETX when using the ring sizes of the Probabilistic key scheme experiments obtained in Chapter 4.

Another essential factor that has a high impact on the increase of the ring sizes is that in the Linear key distribution nodes also consider the voting process to identify honest nodes. This interferes directly with the nodes that might not achieve this requirement and thus are not selected as preferred parent. When nodes are identifying preferred parent in order to compute the best path to reach root nodes regardless of which objective function, each of them go through a voting process with all of its neighbours to vote which node is mostly



(a)
DODAG formation without secure links

(b)
DODAG formation with secure links

Figure 5.10: DODAG formation with and without shared keys distributed to nodes. The connection between all nodes changes in the DODAG when nodes are forced to choose preferred parents with shared keys.

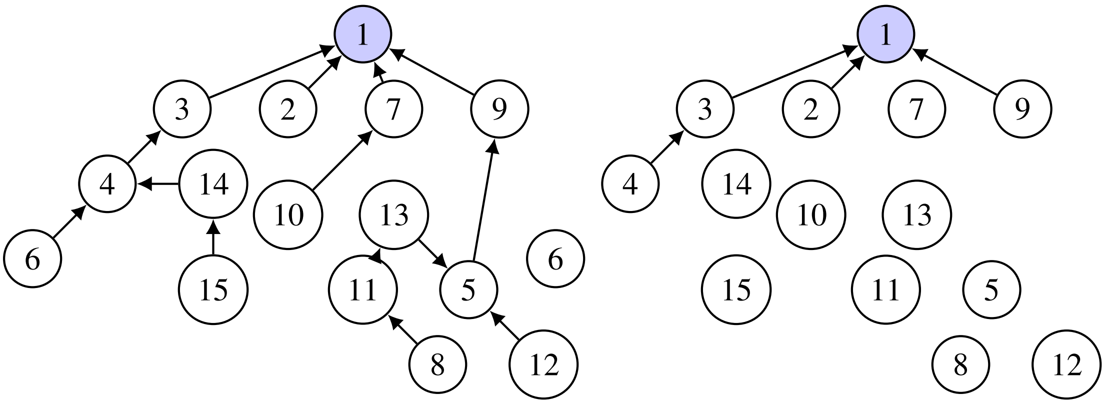
trusted and honest. This by itself discards some nodes to be chosen as preferred parents for others and thus one node will need to find an alternative. This will naturally cause an increase in all ring sizes values.

In the previous Section 5.3 we obtained the number of keys and identifiers needed in each key ring and identifier ring in order to achieve full connectivity when using the Linear key scheme in the content of the IoT running RPL with both OF0 and ETX.

5.4 Fixed Network Experiment

In this section we investigate how the Deterministic scheme perform when running the experiments in the fixed network environment as presented in Section 3.7 for both ETX and OF0 objective functions.

To gain a clear view of why this is happening we experiment on a small network of 15 nodes in a fixed network environment where we distribute a ring size of 8 keys in all nodes following the experiment setup introduced in Section 3.7. We run the experiment for each objective function and without the keys in the nodes. We note that the DODAG formed when using ETX shown in Figure 5.11a has a different topology from the topology of the network when OF0 is used shown in Figure 5.12a. We then load the rings listed in the Appendix C Section 3.2 for nodes 1 to 15. We note that when using the ETX objective function that only 5 nodes join the secure DODAG as shown in Figure 5.11b and 10 nodes are discarded.



(a)
Insecure DODAG using ETX

(b)
Secure DODAG using ETX

Figure 5.11: DODAG using ETX with the DAGs between nodes do not share a key removed from the DODAG in comparison with the insecure DODAG.

We repeat the same experiment with the OF0 objective function shown in Figure 5.12b and we note that when using the OF0 objective function that only 6 nodes join the secure DODAG as shown in Figure 5.11b and 9 nodes are discarded.

We also note for both ETX and OF0 that if one node is discarded from the secured RPL all the leaves of that node are also discarded which is what we expect since for a node to be in the DODAG it needs to have a complete link to reach the root node. We observe that the number of nodes securely connected when using OF0 is more than the ones for the ETX. This is an expected result and it matches with what was expected since the ETX link metric parameter adds a variable constraint when choosing a preferred parent. We also note that the number of nodes securely connected when using the Deterministic scheme in comparison with the Probabilistic scheme results obtained in Chapter 4 Section 4.4

5.5 Evaluation Of The Impact Of Increasing the RS In An IoT Environment

In this section we evaluate the impact of the rings size identified in Section 5.3 to the performance of the network and the RPL DODAG formation. We also investigate the overhead of the Deterministic scheme on the IoT devices used and the network in general.

5.5.1 RPL control messages Number

In this section we evaluate the average number of the RPL control traffic overhead for all nodes. We note in Figure 5.13 and Figure 5.14 that the number of RPL control message increases when the network size increases. We also observe that the number of DAO messages for both ETX and OF0 are a lot larger than the number of DIS and DIO messages. This is obviously the result of how RPL works and the downward direction of DIS and DIO versus the upward direction for the DAO messages that is essential for the calculation of the path to reach the root node. We also note that for all control messages types, the number of messages when using OF0 is greater than when using ETX.

We also note that the number of control messages is not directly related to the pool size but related indirectly to the ring size since the aim is to obtain a secure DODAG and thus nodes sharing a key is an important factor that will impact how many control messages each node will need in order to determine the best path and preferred parent.

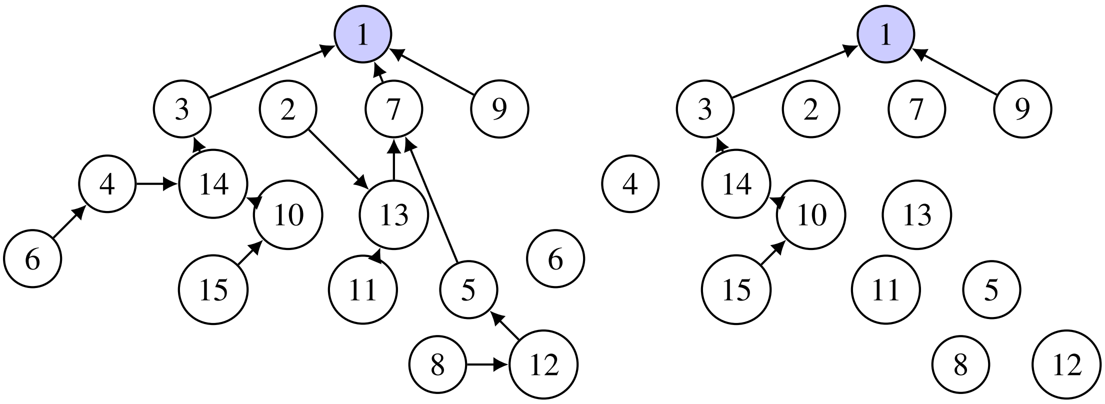
We observe in Figure 5.15 that ETX outperforms OF0 in all RPL control messages as the less the number of RPL control messages there is, the less overhead we are adding which in turn naturally leads to less power consumption. The increase in the total number of control messages for both ETX and OF0 is also related to the voting process and how it impacts the decision of choosing a preferred parent.

Similarly to the increase in the number of RPL control message we observe in Figure 5.16 that the packets delivery ratio decreases when using linear key distribution. This is related to the increase in the number of RPL control messages that are discarded because of no trust when using this scheme. As nodes vote for trusted and honest neighbours increases the number of control messages that will not be refused by the nodes will also increase. This is considered a failed packet similar to the collision of packets that increase with the increase of the number of nodes.

5.5.2 Hops Count Average

In this section we compare the average hops count when using both ETX and OF0. We look at the initial hops count in the first five minutes in comparison with the hops count after RPL converges. Since OF0 selects the suitable path based on the number of hops to reach the root node we expect OF0 to outperform ETX. We consider an objective function outperforming another when the number of hops count is smaller than the other.

We observe the number of hops when using ETX in Figure 5.17 and in Figure 5.18. We note that the average hops count when using OF0 both in the initial stages or when RPL converges is smaller than the hops count for ETX. This is because OF0 computes the shortest path as the best path as ETX does not take the hops count into consideration and looks at other metrics. We also note that the difference in the number of hops between the initial hops



(a)
Insecure DODAG using OF0

(b)
Secure DODAG using OF0

Figure 5.12: DODAG using OF0 with the DAGs between nodes do not share a key removed from the DODAG in comparison with the insecure DODAG.

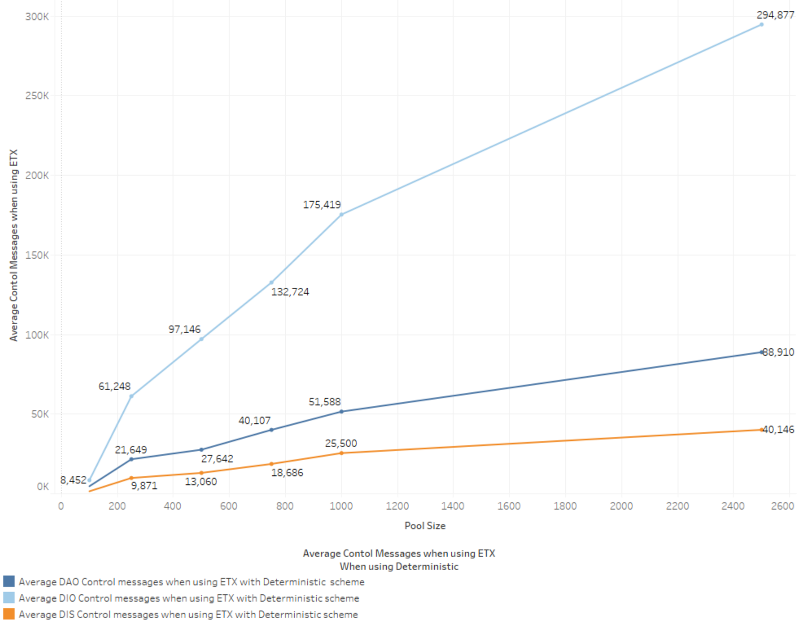


Figure 5.13: Number of RPL Control messages generated in 24 hours for different networks using ETX. All control messages increase when the pool size increases.

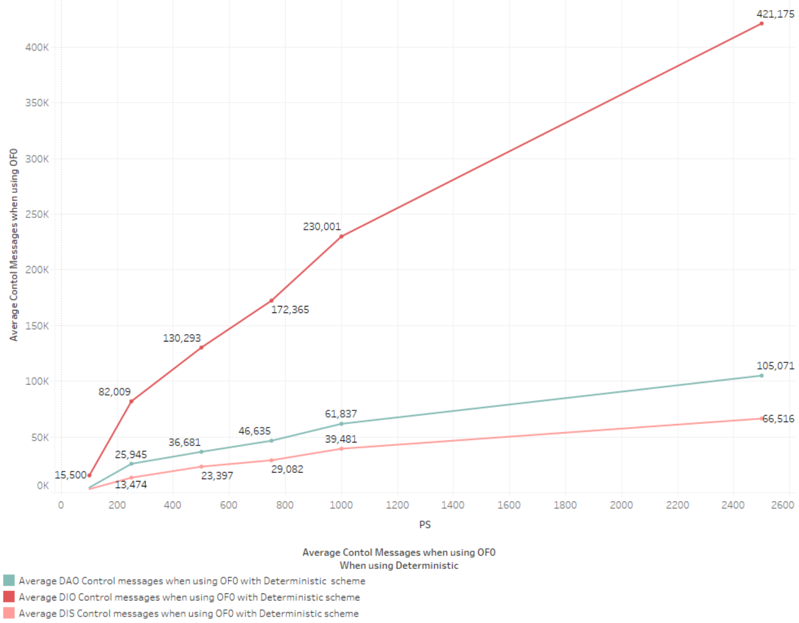


Figure 5.14: Number of RPL Control messages generated in 24 hours for different networks using OF0. All control messages increase when the pool size increases.

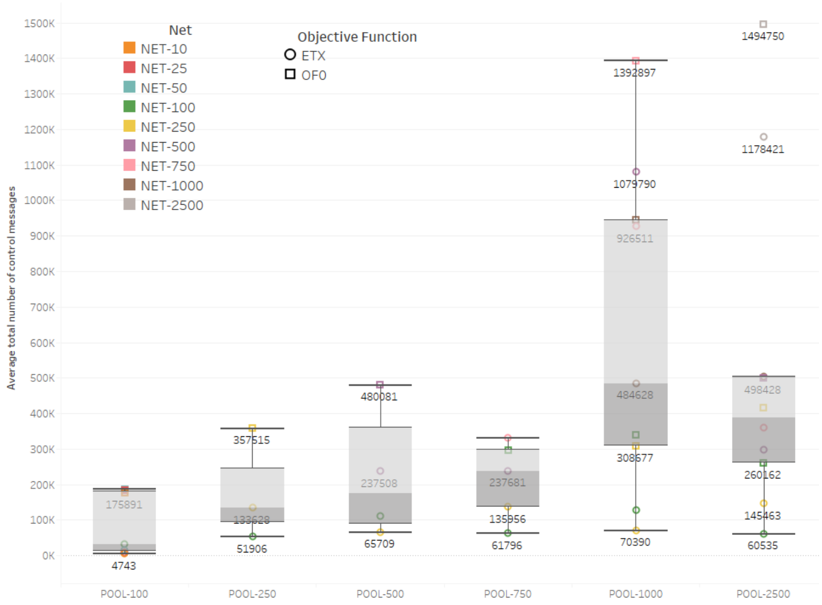


Figure 5.15: Total number of RPL Control messages generated in 24 hours for different networks and pool sizes when using ETX and OF0 with the Deterministic scheme.

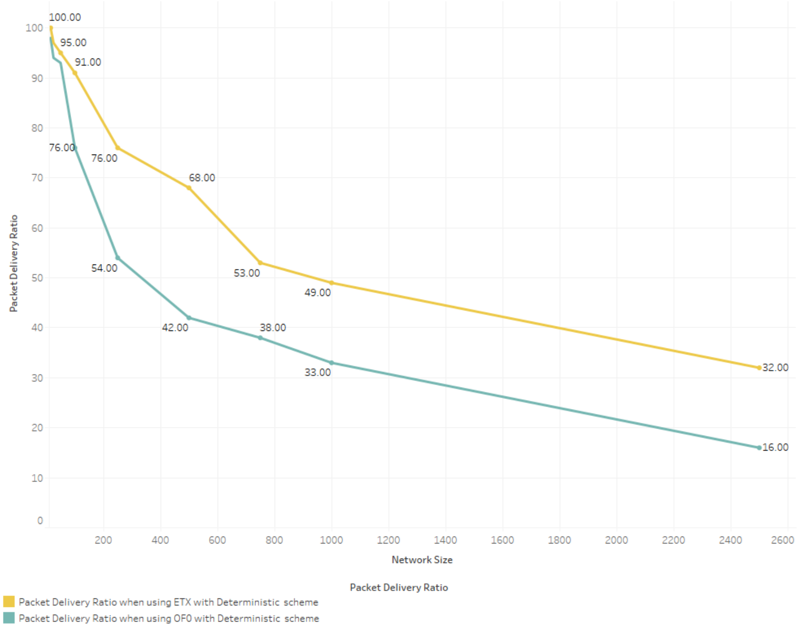


Figure 5.16: Packet Delivery Ratio of control messages for OF0 and ETX. Packet delivery ratio decreases when network increases as the number of control messages increases. ETX outperforms OF0.

count value and the converged one is not too big in ETX but is more apparent in OF0. This is also more clear when the network density is high as it forces nodes to choose a preferred parent without any computation of the node metric and then switch to a lower rank if found. Since the density is high it will take longer to do so as the number of potential preferred parents will be higher.

We also note in Figure 5.19 the average initial hops count when using ETX is relatively larger than the one for OF0 but this difference disappears when the DODAG converges.

5.5.3 Power Consumption

In this section, we look at the average power consumption when using ETX or OF0. Power consumption is collected similarly by using Powertrace and the Energest tool. We look at the power consumption when the radio component is transmitting (Tx), receiving (Rx), the low power mode (LPM) state and the power consumption when the CPU is used in the same fashion as in previous Chapter. We note in Figures 5.21 and 5.20 that there is no significant difference between ETX and OF0 and the average power consumption of all states together is slightly higher for OF. This is more obvious in large networks where the density is high which naturally means there are more control messages being generated in order to

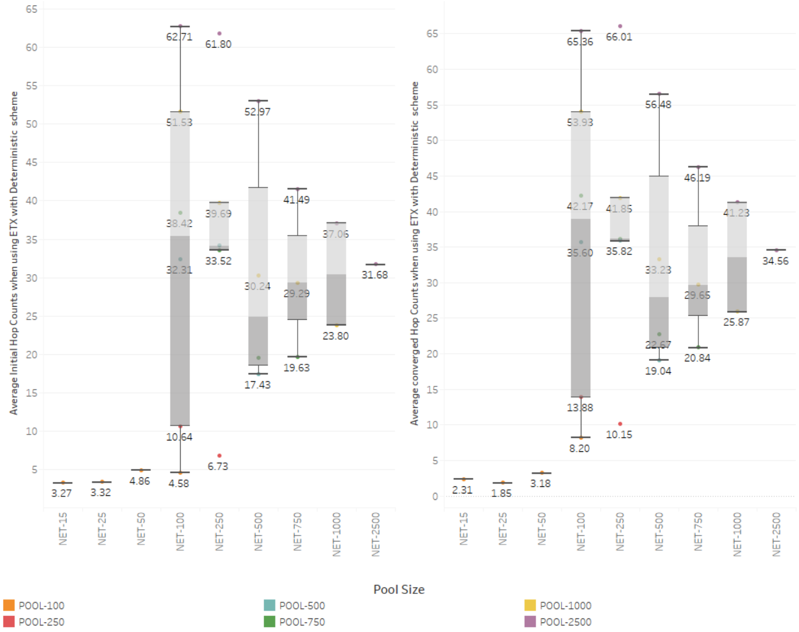


Figure 5.17: Initial and converged average hops count using ETX. Hops count increases when the pool size increases.

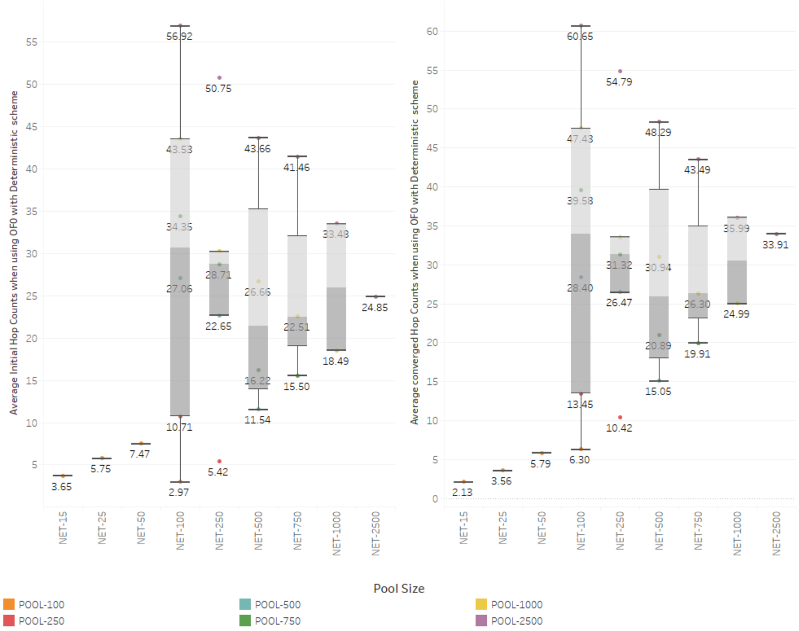


Figure 5.18: Initial and converged average hops count using OF0. Hops count increases when the pool size increases.

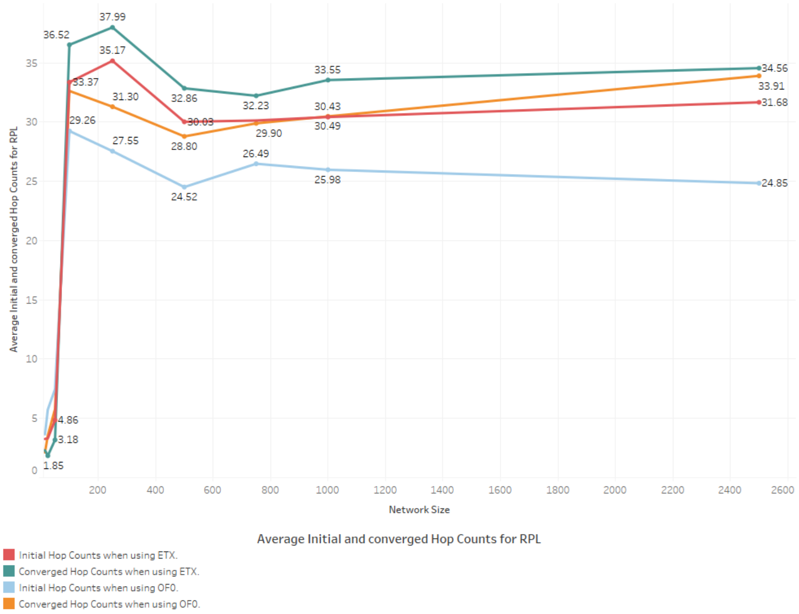


Figure 5.19: Comparison of the initial and the converged average hops count for OF0 and ETX.

identify the preferred parent. The number of neighbours is a more important factor for OF0 as it computes preferred parent and best path based on the link quality and the number of hops.

Looking at the power consumption of the different states in Figures 5.23 and 5.22 we observe that the transmission state TX and the receiver state Rx increase when the density of the network increases but LPM and CPU power consumption do not show any significant changes. We also note that the power consumption for the transmitter state is smaller than the power consumption of the receiver state. This is because the receiver state will stay on for a longer time as the number of neighbours increase.

Finally we compare in Figure 5.24 the average radio duty cycle with the average power consumption for all nodes. We note that there is a significant difference to previous experiments simply because the power consumption increased significantly between the Deterministic key scheme and the Probabilistic key scheme.

5.5.4 CPU Usage

In this section, we evaluate the CPU usage both when the RPL DODAG is forming and once it converges. As observed in Chapter 4, we observe in Figure 5.25 and Figure 5.26 that the CPU usage increases when the network increases. This is expected since an increase in

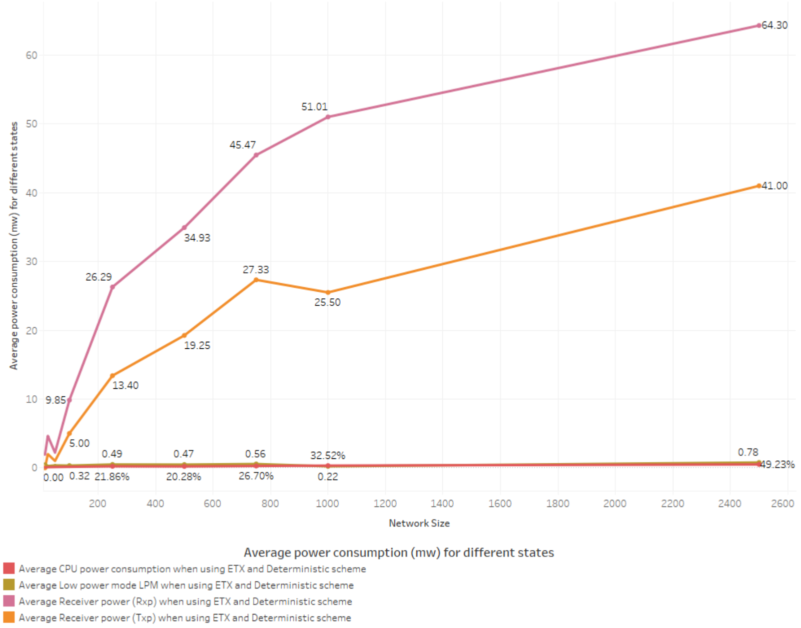


Figure 5.20: States power consumption when using ETX. States power for Transmitting and receiving increases when the network size increase. No significant changes in the low power mode state and CPU power usage.

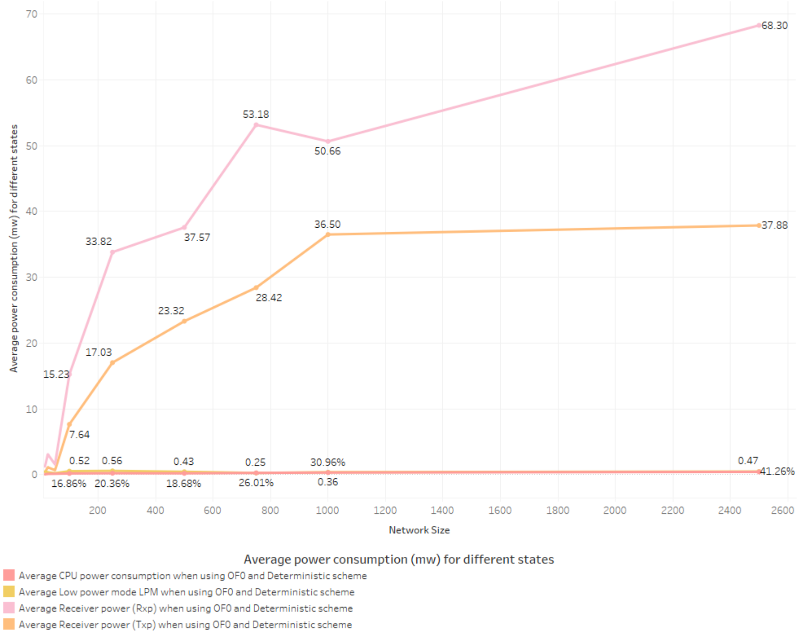


Figure 5.21: States power consumption when using OF0. States power for Transmitting and receiving increases when the network size increase. No significant changes in the low power mode state and CPU power usage.

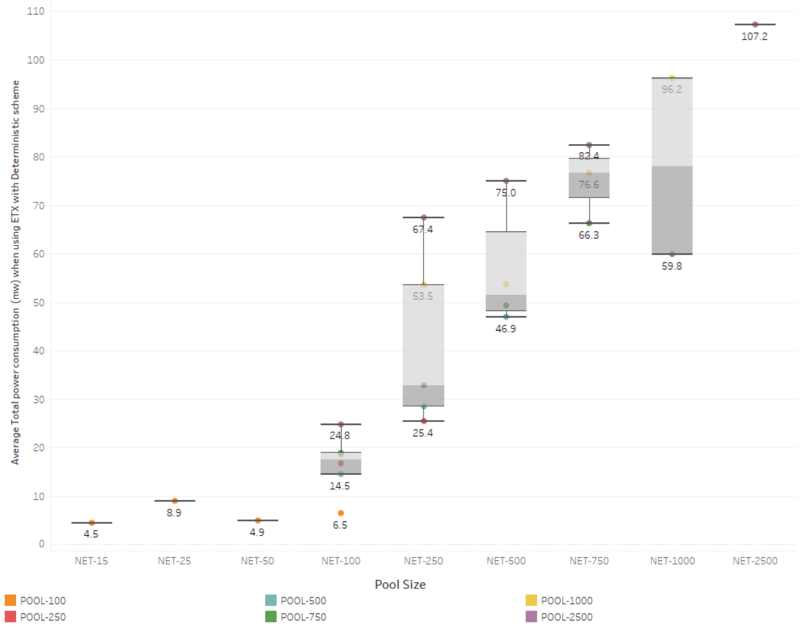


Figure 5.22: Average power consumption for ETX. Power consumption increases when the network size increase due to the increase in the number of messages (increase in network size) and increase in computation for path due to larger pools.

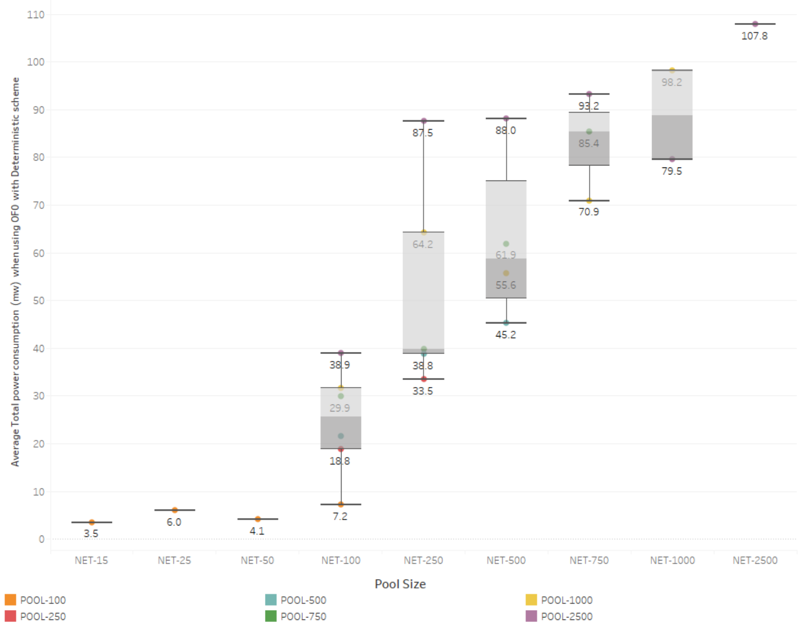


Figure 5.23: Average power consumption for OF0. Power consumption increases when the network size increase due to the increase in the number of messages (increase in network size) and increase in computation for path due to larger pools.

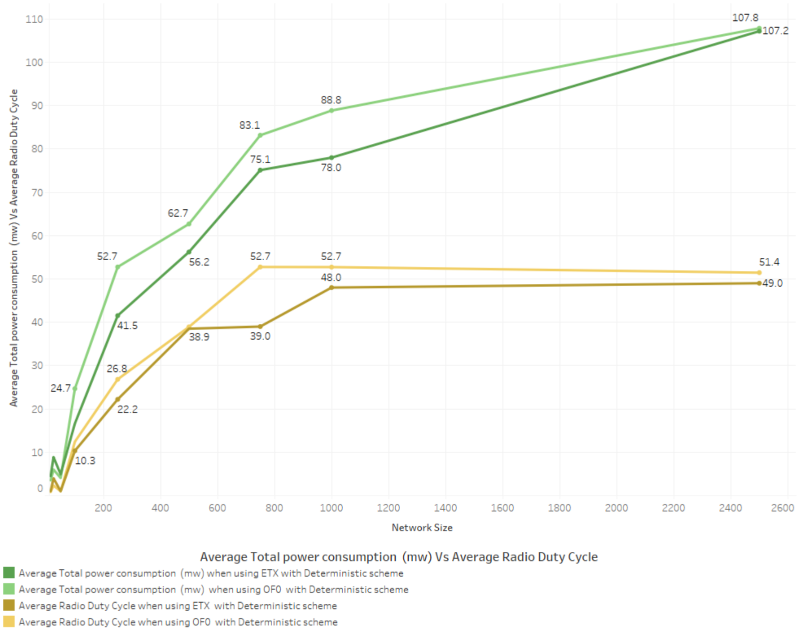


Figure 5.24: Average power consumption vs average radio duty cycle for OF0 and ETX. Of0 underperforms ETX for both the total power consumption and radio duty cycles since the voting process and mutual authentication results in some nodes being discarded.

the number of nodes will force nodes to generate more RPL control messages in order to identify a preferred parent. We also identify that both the initial CPU usage and the converged one do not relate to the ring size.

We also note in Figure 5.27 that OF0 outperforms ETX in terms of CPU usage both for the average initial CPU usage and the average converged CPU Usage. This is because in ETX objective function each node computes the suitable preferred parent by looking at several metrics for all nodes within its range it shares a key with and is trusted in the Linear key scheme. This adds more overhead on the CPU usage in comparison with OF0 that only rely on the link metric and the rank in the network to choose a suitable parent.

5.5.5 Time to Converge

In this section we investigate the time it takes for the DODAG to converge when using both ETX and OF0. We follow the same procedure to obtain the time to converge variables as we did in the previous Chapter and we run all experiments for 24 hours to ensure that no significant changes in the RPL occurs. We observe in Figure 5.28 and Figure 5.29 that ETX and OF0 takes nearly the same time to converge although the number of hops in OF0 is greater. This is due to the fact that the OF0 takes more time to due to the computation of

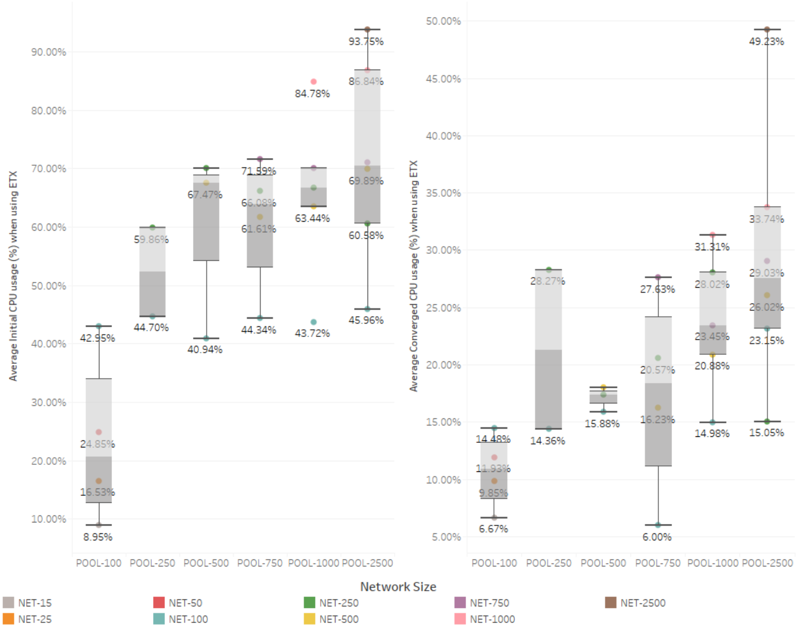


Figure 5.25: Initial CPU usage and converged CPU usage for RPL using ETX for different pools size. The CPU usage increases when the pool size and the network size increase. The CPU usage decreases after the DODAG converges.

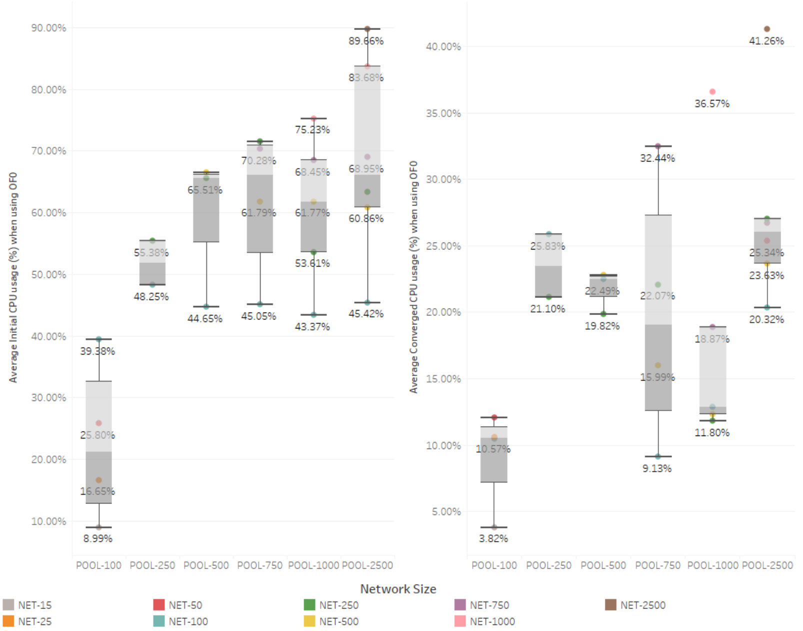


Figure 5.26: Initial CPU usage and converged CPU usage for RPL using OF0 for different pools size. The CPU usage increases when the pool size and the network size increase. The CPU usage decreases after the DODAG converges.

the suitable link while ETX takes time to determine suitable a node by comparing the node metrics of all neighbours of one specific node.

We can see in Figure 5.30 that the time to converge for ETX and OF0 is not really comparable as it is less than a second for large networks and goes to less than half a second for smaller networks.

5.5.6 Latency

In this section we investigate the time it takes for packets to reach the root node in a network. We observe in Figure 5.31 that the latency increases when we increase the number of nodes. We can also identify that latency increases significantly from small networks to networks larger than 100 nodes. This increase is related directly to the number of hops in a DAG as more hops in the DAG means it will take more time for a packet to reach its root node. We also observe that the latency when using ETX is slightly smaller than when using OF0 for large networks. This result is consistent with what was obtained in Chapter 4 in terms of the latency differences between ETX and OF0.

5.5.7 Number of Neighbours

In this section we look at the average number of neighbours in different networks. The average number of neighbours NNE is related to the size of the environment and how many nodes are being simulated. All networks are generated in a random way and locations of each node in the environment is not related to the scheme used. When looking at the number of neighbours in Section 5.32, we observe that the number of secure neighbours decrease when using the Deterministic experiments due to the voting process that forces some neighbours to be discarded in comparison to the Probabilistic scheme. The difference in the number of neighbours when using ETX and OF0 is also consistent with previous results in Chapter 4 as ETX outperforms OF0, however, the difference is larger than the difference when using the Probabilistic scheme.

5.6 Summary

The performance of the linear key pre distributed scheme investigated by [47] was simulated in the context of the IoT following the experiments used in previous experiments in Chapter 4. We identified that the linear scheme underperforms the Probabilistic scheme since the network does not achieve full secure connectivity when using the ring sizes computed in Equation 3.5.1 as explained in Section 5.1 or when obtained in Chapter 4 shown in

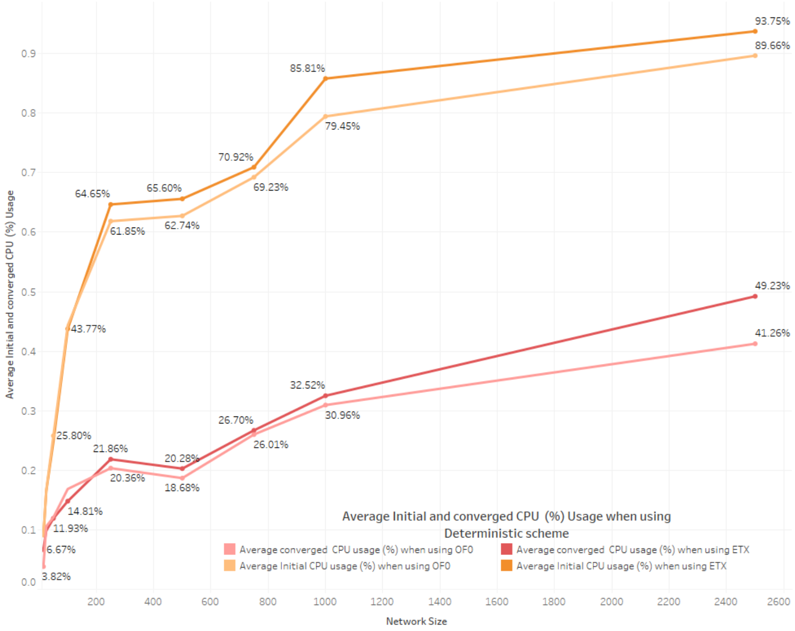


Figure 5.27: Initial CPU usage and converged CPU usage for RPL using ETX vs OF0. The percentage of CPU usage when using ETX outperforms OF0 since ETX computation for the path metric and the ETX cost of the path adds an overhead to the CPU usage.

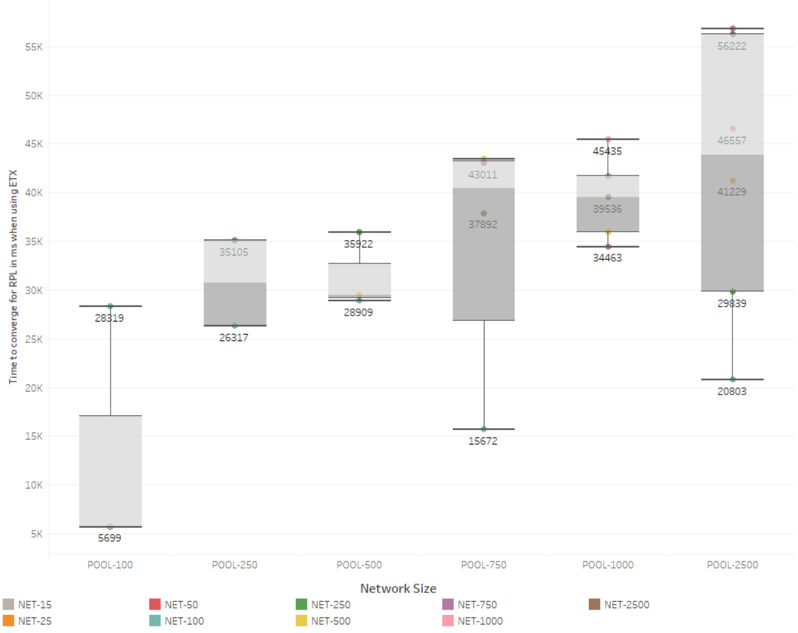


Figure 5.28: RPL DODAG time to converge when using ETX for different pools size. The time to converge increases when network size increases and pool size increases.

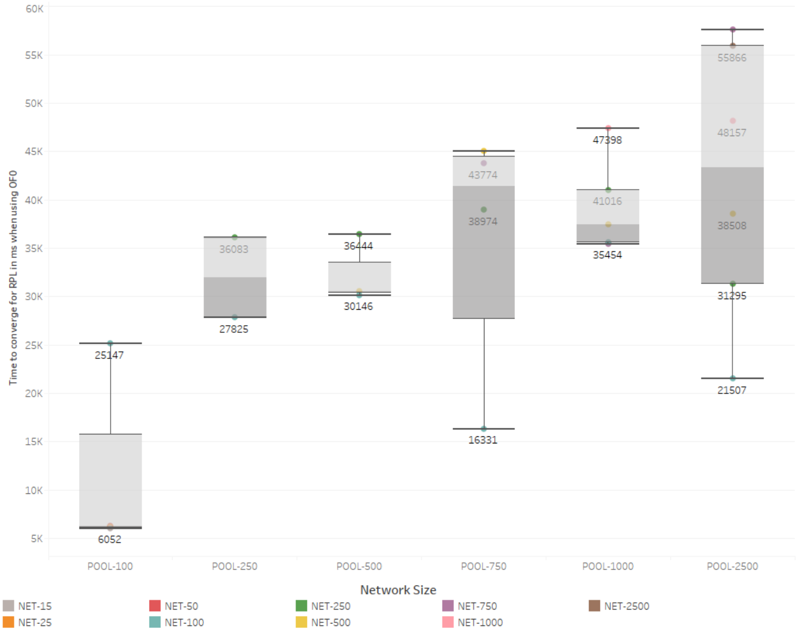


Figure 5.29: RPL DODAG time to converge when using OF0 for different pools size. The time to converge increases when network size increases and pool size increases.

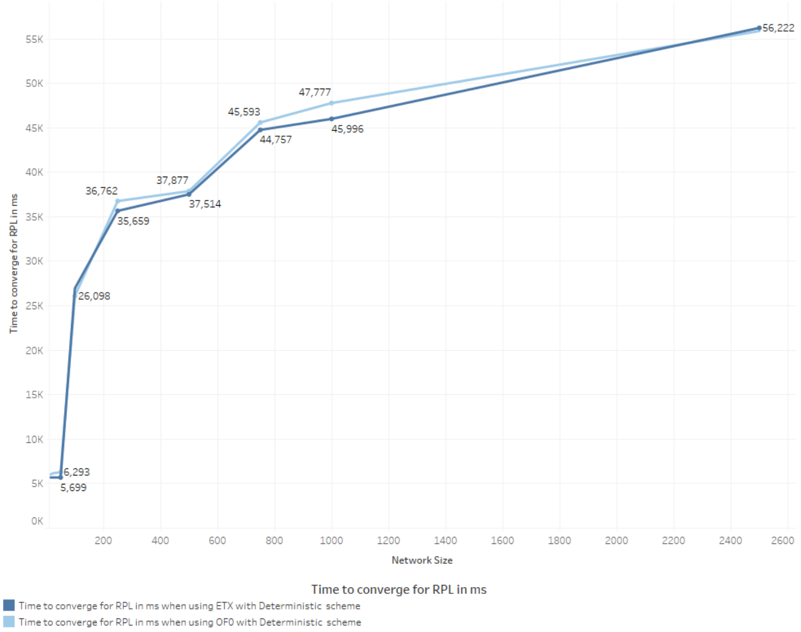


Figure 5.30: RPL DODAG average time to converge for both ETX and OF0 in different networks. The time to converge for all network sizes is similar for both objective functions and there is no significant difference.

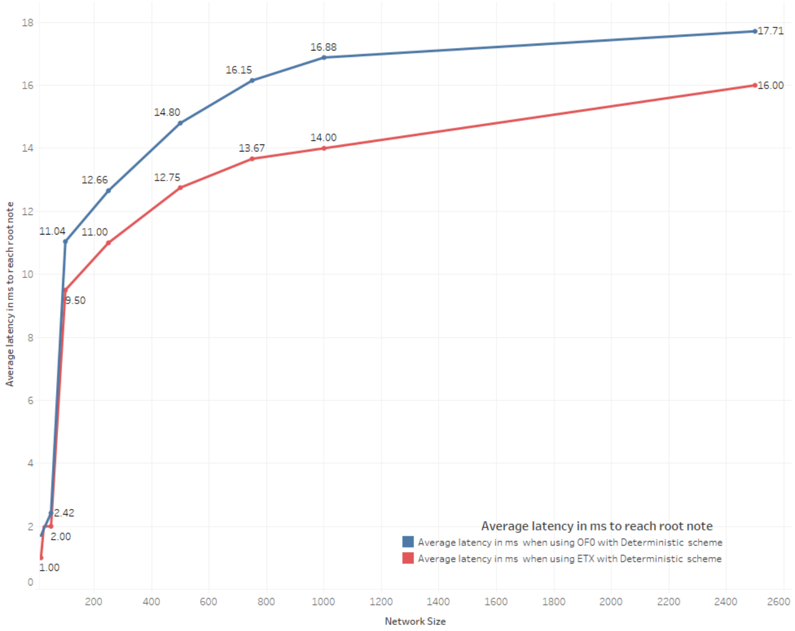


Figure 5.31: Average latency in ms when using OF0 and ETX. OF0 outperforms ETX as the average number of hops to reach the root node for ETX is larger than for OF0 and this in term increases the latency for packets to reach root node.

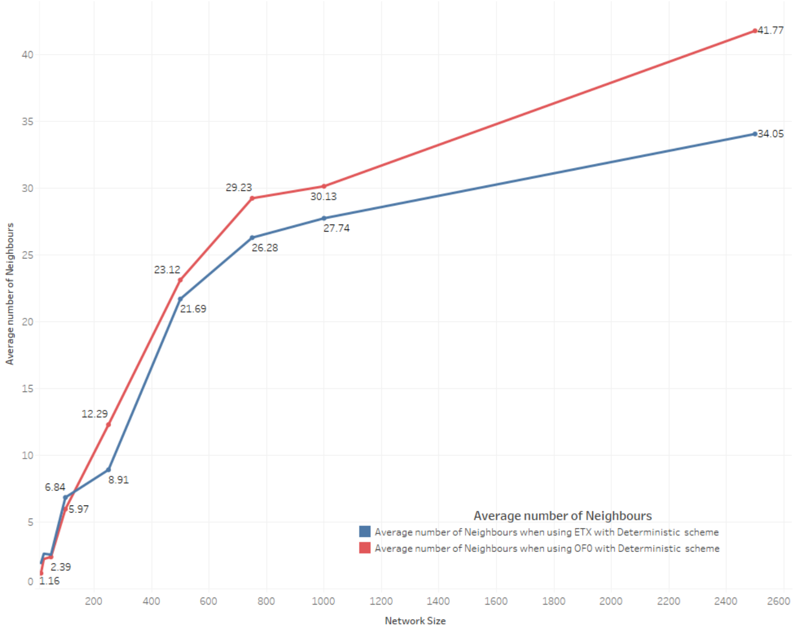


Figure 5.32: Average number of Neighbours when using OF0 and ETX. OF0 outperforms ETX in the average number of neighbours because of the FMAP mutual authentication and the voting process and some nodes are discarded.

Section 5.2. After we evaluated the performance of the scheme in the context of the IoT and determined that the ring sizes obtained in Section 5.2 were not enough we increased the ring sizes until we achieved full connectivity of all nodes in the DODAG in Section 5.3. Finally we have identified that the impact of increasing the ring sizes for different networks and different pools on both the link metrics of the DODAG and the nodes metrics have a negative impact on the link metrics specifically but a positive impact on the nodes metrics. This is due to the fact that the voting process of the linear scheme results in a decrease in the number of neighbours and hence the latency and the hop count increase. The decrease in the number of neighbouring nodes also results in a decrease in power usage and CPU usage as each node has less nodes to communicate with.

Chapter 6

Shared Identifier Secure Link Objective Function (SISLOF)

In this chapter we propose the Shared Identifier Secure Link Objective Function (SISLOF) to find secure links (those that share an identifier) between any node and all of its candidate parents to form a secure RPL routing table while minimising the number of nodes that are excluded because of insecure links. We first define the SISLOF algorithm in Section 6.1. We will also experiment with SISLOF using the Probabilistic scheme as in Chapter 4 in Section 6.2 and the Deterministic scheme as in Chapter 5 in Section 6.3.

Shared Identifier Secure Link Objective Function will attempt to find shared keys between nodes by using the Key pre-distribution algorithm for Distributed Sensor Networks proposed in [46]. This will allow the formation of an RPL routing table that only contains secured links between nodes.

The aim of the Shared Identifier Secure Link Objective Function (SISLOF) is to create a secure RPL routing table with as many nodes as possible. Specifically, its objectives are:

- Only nodes that share a key can become a leaf in the DODAG tree
- Nodes that do not share a key with their selected parent will discard this selection and try to form a leaf with one of the other nodes that received its DIO (Neighbouring nodes)
- If one node share a key with two or more nodes, it will select the preferred parent with the node that has a better transmission count TC in order to form the leaf between the two nodes.

6.1 SISLOF Algorithm

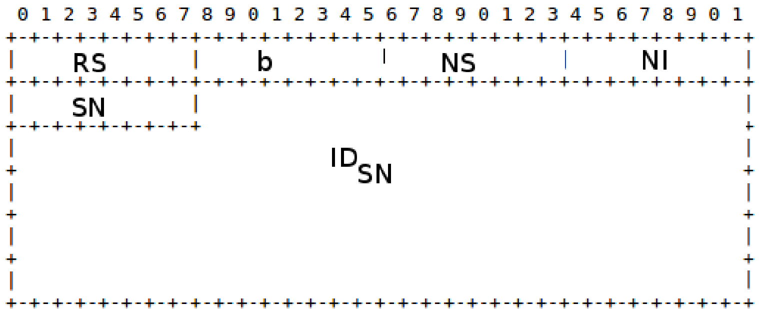
In this section, we explain the SISLOF objective function algorithm that shows how the proposed objective function modifies the messages to incorporate the shared identifiers and shared keys to select a preferred parent. The new objective function is based on the modification of the objective function OF0 based on the ETX objective function parameter of the transmission count TC. The main new modification is the inclusion of the new parameter, Shared Identifier State SIS that defines whether two nodes share one or more identifiers or not. Once this parameter is set to True of false for each neighbouring node, SISLOF will compare the transmission count TC for the link path metric for each of the neighbours to select the preferred parent.

SISLOF uses two types of metrics in its process to compute the preferred parent for a node. First it uses our new node metric object called "**Shared Identifiers State (SIS)**" to compare two arrays of identifiers in order to determine if one or more identifiers exist. This metric is an additive metric since it only reports Boolean values of true or false. It is given the 'A' field value of zero as per IANA codespace for Routing Metrics/Constraints of Common Header 'A' Field [66].

If the node that received the DIO identifies that it shares one or more identifier with two or more nodes, the node will need to choose which of those nodes that sent the DIO will be selected as preferred parent. SISLOF will then need to choose which node that it shares a key with will be chosen as a preferred parent. This will require SISLOF to use a link metric object as a second criteria in order to select its preferred parent. SISLOF will use the TC Reliability object to select the preferred parent. The TC value was calculated for each link that it received a DIO message from and identified that is shares one or more identifiers with. The node that has the lowest TC value will be selected as the preferred parent. The TC is number of transmissions the node expect to make to a destination in order to successfully deliver the packet. This will also require changing the 'A' field of the header 'A' field to 7 for each message, a field given to indicate that the header will report a minimum or a maximum.

6.1.1 Message & Modifications

The SISLOF Objective function will require the modification of the DIO and DAO RPL messages in order to encapsulate the various variables of SISLOF required to exchange identifier rings and look for a common one. Those variables will be either encapsulated in the DIO message sent to a node or in the DAO message replying. Those variables are explained in Table 6.1.



```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      RS       |       b       |      NS       |      NI       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      SN       |                                               |
+-+-+-+-+-+-+-+-+                                               +
|                                                               |
+                             ID_SN                             +
|                                                               |
+                                                               +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6.1: Security fields modifications to the DIO message. These includes the identifiers ring transmission configuration options for SISLOF.

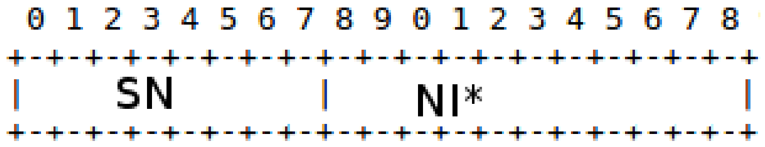
SISLOF variables shown in Figure 6.2 are composed mainly of identifiers and other values related to the segmentation of those identifiers. To incorporate the SISLOF variables shown in Table 6.1 in a DIO message, the 6LoWPAN message, the ICMPv6 control message and the DIO base object requires 89 bytes which implies that there are 38 bytes in the data frame to be used to embed in frame variables related to SISLOF objective function. In Figure 6.2 RS and b are selected to fulfil requirements of the algorithm of [46]. NI provides the number of identifiers that can fit in the DIO payload. NI is calculated as the rounded integer of the available payload (33 bytes) by the identifier size b. NS is the total number of messages required to transmit the complete identifier ring. NS is calculated as the quotient of RS divided by NI. Finally SN identifies the order of the specific message in the complete sequence of messages required to disseminate the identifier ring. It is calculated as the sequence index corresponding to the current message.

Variable	Name of Field	Size in bytes
RS	Ring Size	1 byte
b	Identifier Size	1 byte
NI	Number of identifiers in one message	1 byte
NS	The Total Number of Sequences	1 byte
SN	The Sequence Number	1 byte

Table 6.1: Identifier transmission configuration options for SISLOF. Security variables encapsulated in the DIO message sent to a node or in the DAO message replying.

To encapsulate as many identifiers as possible in each DIO message, variables size in bytes are kept to the minimum by giving only 1 byte for each variable as shown in Table 6.1. This means that each variable can have any value between 0 to 255 in decimal. Several factors were behind choosing these values. From experiments we did and using the same technique used in [46] with a 2500 nodes network and the Ring Size RS that we used was 41 keys/identifiers for each ring. Using the same formula in [46] with the same network size and Pool size, the ring size for a network of 100 000 nodes will be 250 keys. It can be represented in a 1 byte field. We have also used an Identifier Size of 1 byte. 1 byte for the Identifiers is more than enough, since the identifier is not used to encrypt the message and it is only used to identify if a common key exist between two nodes. Using both RS and b will not yield a number of identifiers in one message larger than 256. In our example and if using the same number of nodes as [46] will yield one identifier NI per each message and that is 250 messages or the total number of sequences NS. The sequence number SN will of course be smaller than NS as it is a counter that will determine the sequence number of a specific message.

The 6LoWPAN message shown in Section 2.4.1 takes 69 bytes message which leaves us with 58 bytes in the data frame that we used to embed frames related to our Objective Function. Our proposed Objective function used the 58 bytes of the DAO as below and shown in Figure 6.2.



```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    SN        |    NI*                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6.2: Security fields modifications to the DAO message header. These includes the identifiers ring transmission response for common identifiers check for SISLOF.

SN is the sequence number received in the corresponding DIO. NI is a bitmap with bits set to 1 if the identifier with the corresponding position is available in the identifier ring of the node that received the DIO message and 0 otherwise. [1].

6.1.2 Securing The Link

A node that is propagating the DODAG information, broadcasts the DIO message downwards. The DIO message will contain as in Figure 6.1 all the information related to 6LoWPAN messages such as the IPv6 header, etc. On top of this, the DIO message will also contain its rank with the root. The SISLOF objective function addition to the DIO message, explained in Figure 6.1 will also contain the identifiers of the first DIO frame from the sequence of frames (SN).

[1] NI size is variable and changes depending on the size of each identifier

One of the constraint variables that is required by the SISLOF objective function is the shared identifier constraint. The calculation of this variable will produce a secure or insecure link. This variable will determine whether a node is considered a secure candidate parent or not. All other variables are discussed at a later stage in Section 6.1.3. This is the first constraint that SISLOF computes before moving to other variables to calculate the path between nodes and the root and form the RPL routing table. The first stage involves DIO messages broadcast downward by each node that is part of an RPL DODAG downward as shown in Algorithm 1 below:

Each node that receives a DIO message replies back with the DAO message that contains as of Figure 6.1, all information related to 6LoWPAN message such as the IPv6 header, etc. Further more, the DAO message will also contain the SISLOF objective function additions explained in Figure 6.2. The DAO messages sent upward by each node that received the DIO is shown in Algorithm 2 below:

The sequence diagram shown in Figure 6.3 shows the various control messages and variables exchanged between two nodes in order to determine if a common identifier exists. After a common identifier is found, SISLOF will then compute the link metrics and the parent TC in order to choose the preferred parent.

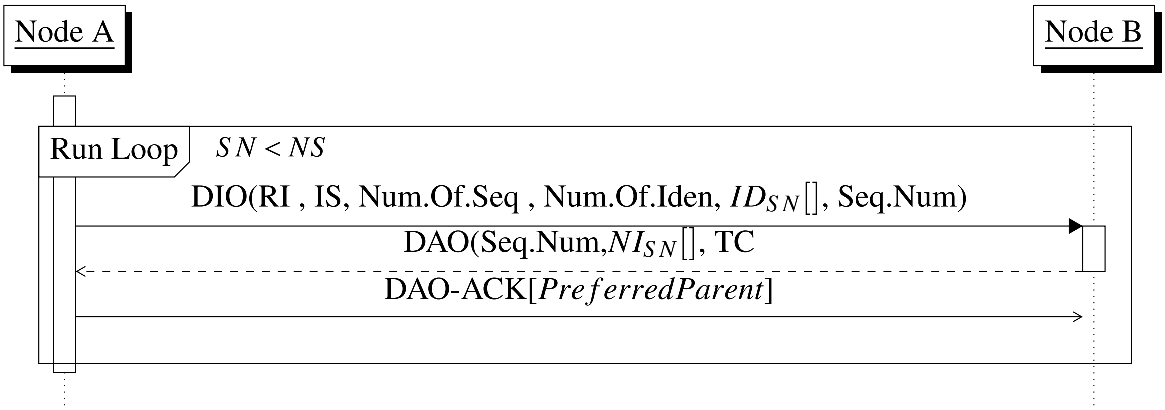

Node A
Node B
Run Loop
$SN < NS$
DIO(RI , IS, Num.Of.Seq , Num.Of.Iden, $ID_{SN}[]$, Seq.Num)
DAO(Seq.Num,$NI_{SN}[]$, TC
DAO-ACK[$PreferredParent$]

Figure 6.3: SISLOF Sequence Diagram showing the security variables in DIO messages and the response to common identifier in the DAO message fields.

6.1.3 Link Metrics & parent TC Calculation.

If one or more secure node that received the DIO identifies that a shared identifier exist then the expected Transmission Count metric (TC of the parent), similarly to the TC calculation of RPL link metrics in Section 2.4.2, the metric will become the second criteria on deciding the best parent. This metric will return the values of the DIO origin node TC

Algorithm 1: DIO Messages Algorithm showing how message is broadcasted downward by each node that is part of an RPL DODAG downward.

```
Input  : Ring Size $RS$, Identifier Size $b$, Number of Bytes available in frame in
         bits $B$, Identifier Ring of Sender $IR_s$
             $IR_s = \left[ ID_1,\ ID_2,\ ID_3,\ ID_4,\ \ldots)\ ID_{(n-1)},\ ID_n \right]$
Output : Identifier Ring for each frame $IR_{SN}$
             $IR_{SN} = \left[ ID_1,\ ID_2,\ ID_3,\ ID_4,\ \ldots\ ID_{(NI-1)},\ ID_{NI} \right]$
         DIO message $DIO_{SN}$=($n$, $b$, $IR_{SN}$, $NI$, $NS$, $SN$).
Calculate number of Identifiers in a frame $NI$
    $NI = Integer(\frac{B-40}{b})$
  Calculate Number of sequences NS
    $NS = RoundToLargestValue(\frac{RS}{NI})$
$SN = 1$;
$x = 0$;
for $SN$ to $NS$ do
    $IR_{SN} = [x]$;
    $y = (NI * SN) - 1$;
    for $x$ to $IR[y]$ do
        Append $IR_s[x]$ To $IR_{SN}$;
        Increment $x$;
    AddtoDictionary $DIO_{SN}$ ( ($RS$) "Ring Size" , ($b$) "Identifier Size" , $NS$ "Total
      Number of Sequences", $NI$ "Number of Identifiers in one frame", $IR_{SN}$
      "Identifier Ring array for each sequence", $SN$ "Sequence Number for each
      frame" );
    Send $DIO_{SN}$ Downward to All nodes ;
    Increment $SN$;
```

Algorithm 2: The algorithm for DAO messages sent upward by each node that received the DIO.

Input :

- **DIO message** (DIO_{SN})

  $$DIO_{SN}=(n, b, IR_{SN}, NI, NS, SN)$$

- **Identifier Ring of Receiver** IR_r

  $$IR_r = \Big[ID_1,\ \ ID_2,\ \ ID_3,\ \ ID_4,\ \ \ldots)\ \ ID_{(n-1)},\ \ ID_n\Big]$$

- **Ring Size (RS)**

Output :

- **Shared identifiers bits** (SIB_{SN})

  $$SIB_{SN} = \Big[b_1,\ \ b_2,\ \ b_3,\ \ b_4,\ \ \ldots\ \ b_{(NI-1)},\ \ b_{(NI)}\Big]$$

- **DAO message**

  $$DAO_{SN}=(SN), SIB_{SN}$$

- **Shared Identifier State** (SIS)

  $$SIS = \Big[b_1,\ \ b_2,\ \ b_3,\ \ b_4,\ \ \ldots\ \ b_{(NI-1)},\ \ b_{(NI)}\Big]$$

```
$SIB_{SN} = [NI]$;
$x = 0$;
$y = 0$;
$z = 0$;
$w = 0$;
$SIS = [w]$;
for $w = 0$ to $RS - 1$ do
    for $y = 0$ to $NI - 1$ do
        for $z = 0$ to $RS - 1$ do
            if $IR_{SN}[y] = IR_r[z]$ then
                Append 0 To $SIB_{SN}$;
                $SIS[w] = 0$ ;
            else
                Append 1 To $SIB_{SN}$;
                $SIS[w] = 1$ ;
AddtoDictionary $DAO_{SN}$ ($SIB_{SN}$ "Shared Identifiers bits", $(SN)$ "Sequence
    Number" );
Send $DAO_{SN}$ upward to DIO Sender ;
```

Pool	Network	Original			Probabilistic scheme				SISLOF
		RS	PSK%	ETX	ETX		OF0		
			OF0	ETX	RS	PSK%	RS	PSK%	RS
100	100	8	89.96%	54.21%	29	100.00%	23	100.00%	12
250	250	13	73.68%	60.30%	47	100.00%	36	100.00%	20
500	500	15	63.10%	67.13%	58	100.00%	48	100.00%	28
750	750	22	70.16%	52.28%	79	100.00%	63	100.00%	38
1000	1000	25	58.97%	65.38%	92	100.00%	77	100.00%	40
2500	2500	41	73.29%	64.58%	115	100.00%	104	100.00%	60

Table 6.2: Ring sizes when using SISLOF for various pools when the network size is the same. The ring sizes to achieve full connectivity when using SISLOF in comparison with Ring sizes for ETX and OF0 when using Probabilistic scheme.

(*parent_metric*) and its received metric *instance_C*. From these two variables the link metric can be calculated to return the TC of the link *link_metric* [102].

6.2 SISLOF With Probabilistic Key Pre-Distribution Scheme

In this section, an implementation of SISLOF using the Probabilistic key distribution scheme as in Chapter 4 is presented. Similarly to the previous Chapters 4 and 5 the generation of Keys Pool, IDs pool, Key rings and ID rings were computed using Equation 3.5.1 from [46] and then increasing the RS values gradually until full connectivity of the network is achieved. This presented us with three different sets of experiments, the first in which the key pre-distribution scheme was simulated in the context of Wireless Sensor Networks using RS values obtained in [46] and SISLOF objective function shown in Section 6.2.1. The second experiment in which the SISLOF objective function is used with the RS values obtained full connectivity of the network when using Probabilistic scheme and computed in Chapter 4 discussed in Section 6.2.2. The third experiment computes the (RS) needed to achieve full connectivity of the network using SISLOF objective function and discussed in section 6.2.3. The number of keys in the ring size RS for each of the three set of experiments is shown in Table 6.2 below. It shows the size of the ring needed to achieve 100% connectivity for each Pool size when the network size is the same.

From Table 6.2, we can notice that the ring sizes in SISLOF are lower when compared to the rings sizes needed to achieve full connectivity when using Probabilistic schemes in Chapter 4. We also observe the performance of the key pre-distribution using the four experiment sets results presented in the table. The key pre-distribution in the DSN networks presented the lowest ring sizes and the IoT using the Minimum ETX metric for RPL showed the highest ring sizes. Wireless Sensor Networks required the smallest ring sizes to achieve full connectivity simply because in DSN a node that does not share a key with one of its neighbours can send data to that specific neighbour indirectly through another node and thus the full network connectivity is achieved even if not all nodes share keys. The ring size needed to achieve full connectivity when RPL was used with its default minimum ETX metric was the largest because only nodes that share a key can participate in the RPL routing table. Nodes that did not share keys could not communicate. By increasing the size of the ring, we ensured in [2] that all nodes can join the RPL routing table and thus communicate.

The ring size increases when the network size increases. This result aligns with results obtained in previous chapters since the number of keys in the pool is larger than the number of nodes in the network, the probability of two nodes sharing a key decreases.

6.2.1 Network Connectivity With RSs Computed with DSN

In this section we investigate the ring sizes for different networks when using various pools. From Figure 6.4, we note that the for different networks using the ring sizes computed in Equation 3.5.1 that the percentage of shared keys is higher when using SISLOF than using either ETX or OF0 as shown in Figures 4.1 and 4.2. This is an expected result as the DAG when using SISLOF objective function is formed only between nodes that share a key in comparison with ETX and OF0. When using ETX Objective, the DAG is formed by computing the preferred parents by comparing the number of transmissions needed to reach the root node. When using OF0 Objective, the DAG is computed by identifying nodes that provides good connectivity without using a specific metric and giving priority to the rank value of the node.

When using either OF0 and ETX experiments, the number of connected nodes decreases as the metrics are not associated with sharing a key and therefore all nodes that do not share a key do not join the DAG. This naturally resulted in a decrease of the number of securely connected nodes in comparison with connected nodes as shown in Figure 6.5 and in the percentage of nodes that share a key as shown in Figure 6.4.

6.2.2 Network Connectivity With RSs For Probabilistic Scheme.

In this section we evaluate the performance of the DAG in terms of the number of nodes that share a key and the number of nodes that are securely connected. It is noticed that the DAG achieves full connectivity for all network sizes evaluated when using various pools. The experiment result means that the ring size needed to achieve full connectivity when

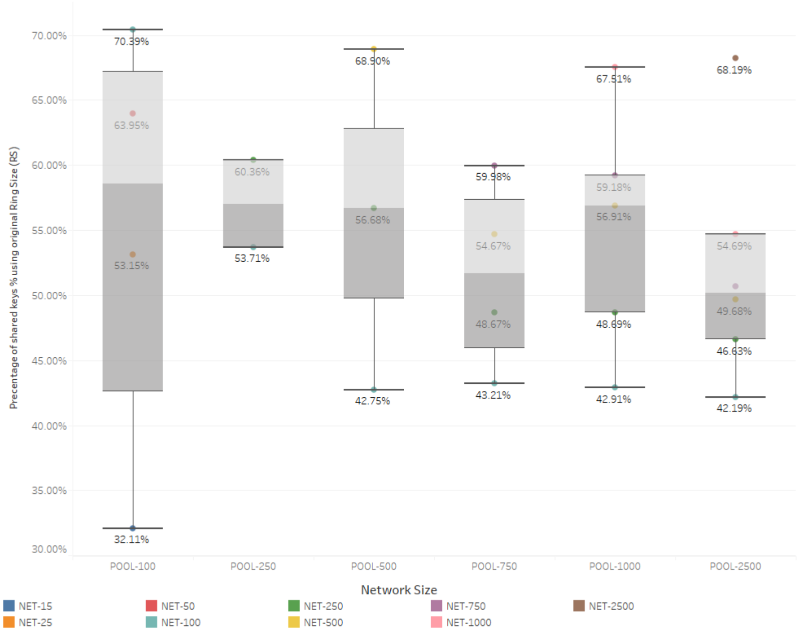


Figure 6.4: Shared key percentage when using *RS* computed in Equation 3.5.1 with Probabilistic scheme. The percentage of shared keys increases when the network size increases for each pool.

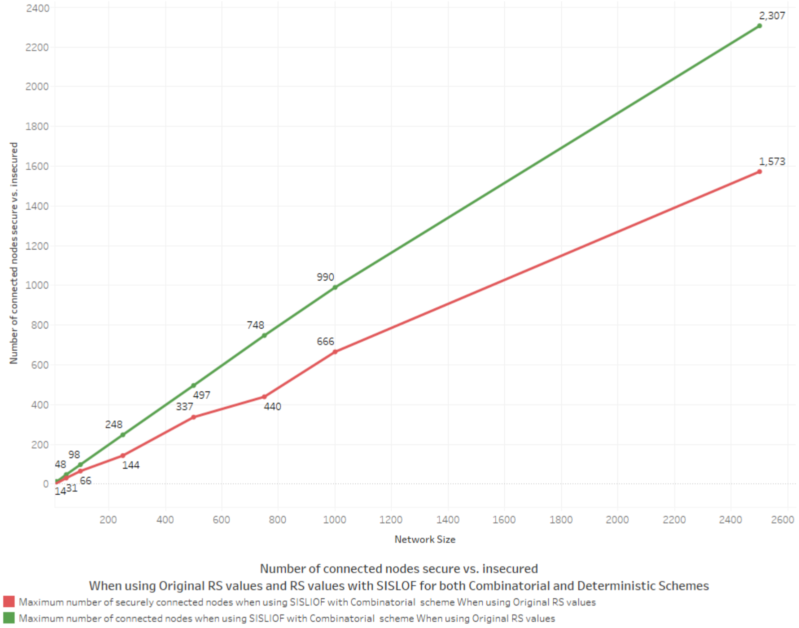


Figure 6.5: Number of DAGs in the DODAG (NNC) vs the number of secured when using *RS* computed in Equation 3.5.1 with Probabilistic scheme.

using SISLOF is smaller than the ring size values obtained in Chapter 4 for both ETX and OF0. In the next section we investigate the minimum ring size values needed to achieve full connectivity of the DAG.

6.2.3 Network Connectivity With IoT RSs

In the previous section we identified that the ring sizes needed when using the Probabilistic scheme for SISLOF is smaller than the ring sizes computed in Chapter 4 . In this section we investigate the ring size needed to achieve full connectivity for each network size when using various pools by starting from the ring size values computed in Equation 3.5.1 as we have already identified in Section 6.2.1 that when using those ring size values the DAG contained a higher number of nodes securely connected, however, it did not reach full connectivity of the network.

We observe in Figure 6.6 that for all experiments for the different network and pool sizes that the ring sizes needed to achieve full connectivity is smaller than the ring sizes needed to achieve full connectivity for both ETX and OF0. We note that the ring size values for all networks when using various pools is nearly half the ring sizes needed for ETX and OF0 and in smaller networks this increases to nearly three times smaller.

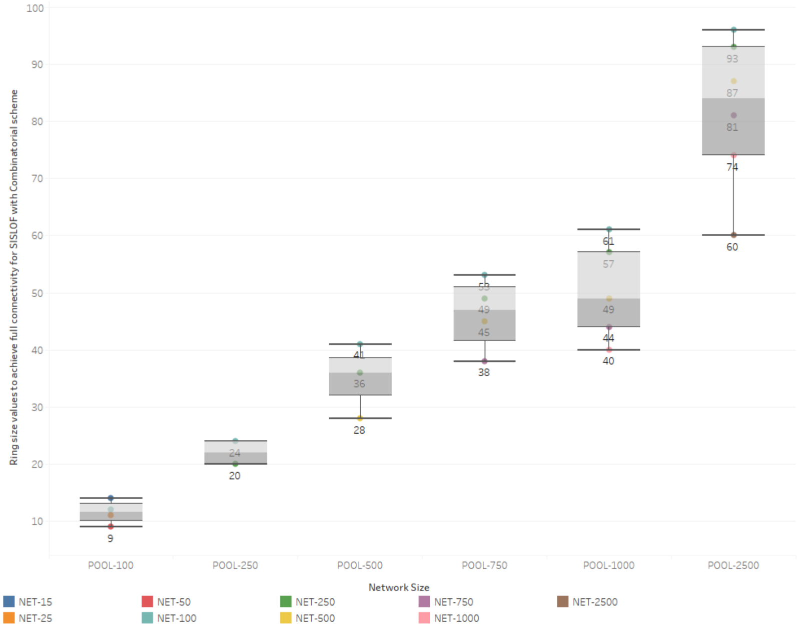


Figure 6.6: Ring size values to achieve full connectivity of network for Probabilistic scheme. Ring size increases when the pool increases and decreases for each pool when the network increases.

6.3 SISLOF With Deterministic Key Pre-Distribution Scheme

In this section, we investigate the performance of SISLOF objective function when using Deterministic key distribution scheme. Similarly to the previous chapters 4 and 5 the generation of Keys Pool, IDs pool, Key rings and ID rings were completed using values from [2].

In this section, an implementation of SISLOF using the Deterministic key distribution scheme as in Chapter 5 is presented. Similarly to the previous Chapters in 4 and 5 the generation of Keys Pool, IDs pool, Key rings and ID rings were computed using Equation 3.5.1 from [46] and then increasing the RS values gradually until full connectivity of the network is achieved. This presented us with three different sets of experiments, the first in which the key pre-distribution scheme was simulated in the context of Wireless Sensor Networks using RS values obtained in [46] and SISLOF objective function shown in Section 6.3.1. The second experiment in which the SISLOF objective function is used with the RS values obtained full connectivity of the network when using Probabilistic scheme and computed in Chapter 5 and as discussed in Section 6.3.2. The third experiment computes the (RS) needed to achieve full connectivity of the network using SISLOF objective function and discussed in Section 6.3.3. The number of keys in the ring size RS for each of the three set of experiments is shown in Table 6.3 below. It shows the size of the ring needed to achieve 100% connectivity for each Pool size when network size is the same.

Pool	Network	Original			Deterministic scheme				SISLOF
		RS	PSK%	ETX	ETX		OF0		
			OF0	ETX	RS	PSK%	RS	PSK%	RS
100	100	8	89.96%	54.21%	21	100.00%	61	100.00%	17
250	250	13	73.68%	60.30%	38	100.00%	49	100.00%	24
500	500	15	63.10%	67.13%	55	100.00%	61	100.00%	32
750	750	22	70.16%	52.28%	73	100.00%	86	100.00%	41
1000	1000	25	58.97%	65.38%	114	100.00%	116	100.00%	45
2500	2500	41	73.29%	64.58%	127	100.00%	136	100.00%	66

Table 6.3: Rings size when using SISLOF for various pools when the network size is the same. The ring sizes to achieve full connectivity when using SISLOF in comparison with Ring sizes for ETX and OF0 when using Deterministic scheme.

We observe in Table 6.3 that all nodes in the networks compared were able to join the DAG securely and this resulted in full connectivity of the network. We also note that percentage of shared keys between nodes when using ring sizes computed in Equation 3.5.1

is higher than when using ETX and OF0 as in Chapter 5 although it did not achieve full connectivity.

6.3.1 Network Connectivity With RSs Computed When Using Probabilistic Scheme

In this section we investigate the ring sizes for different networks when using various pools. From Figure 6.7, we note that for the different networks the ring sizes computed in Equation 3.5.1, however, it is also clear that the percentage of shared keys is higher when using SISLOF rather than using either ETX or OF0 as shown in Figures 5.1 and 5.5. This is an expected result as the DAG when using SISLOF objective function is formed only between nodes that share a key in comparison with ETX and OF0. When using ETX Objective, the DAG is formed by computing the preferred parents by comparing the number of transmissions needed to reach the root node. When using OF0 Objective, the DAG is computed by identifying nodes that provides good connectivity without using a specific metric and giving priority to the rank value of the node.

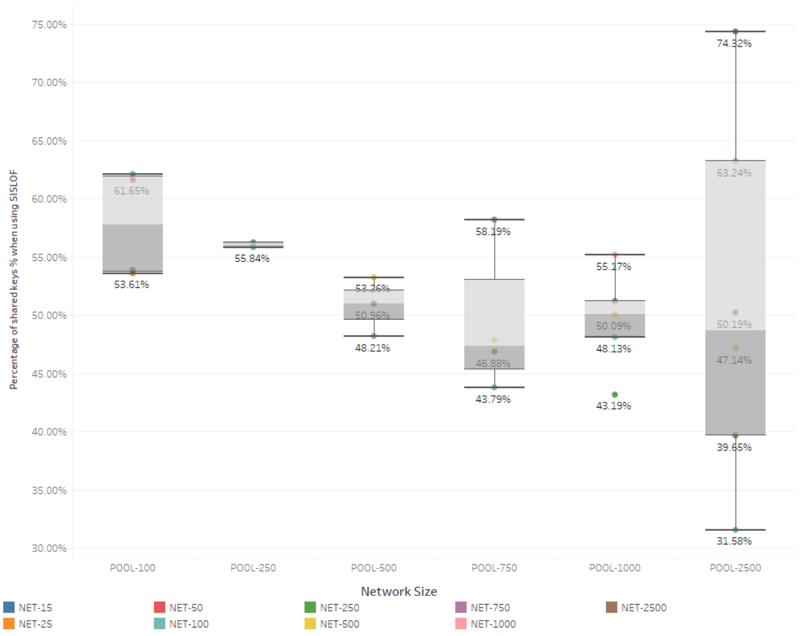


Figure 6.7: Shared key percentage when using *RS* computed in Equation 3.5.1 with Deterministic scheme. The percentage of shared keys increases when the network size increases for each pool.

When using either OF0 and ETX experiments, the number of connected nodes decreases as the metrics are not associated with sharing a key and therefore all nodes that do not share a key do not join the DAG. This naturally resulted in a decrease of the number of securely connected nodes in comparison with connected nodes as shown in Figure 6.8 and in the percentage of nodes that share a key as shown in Figure 6.7.

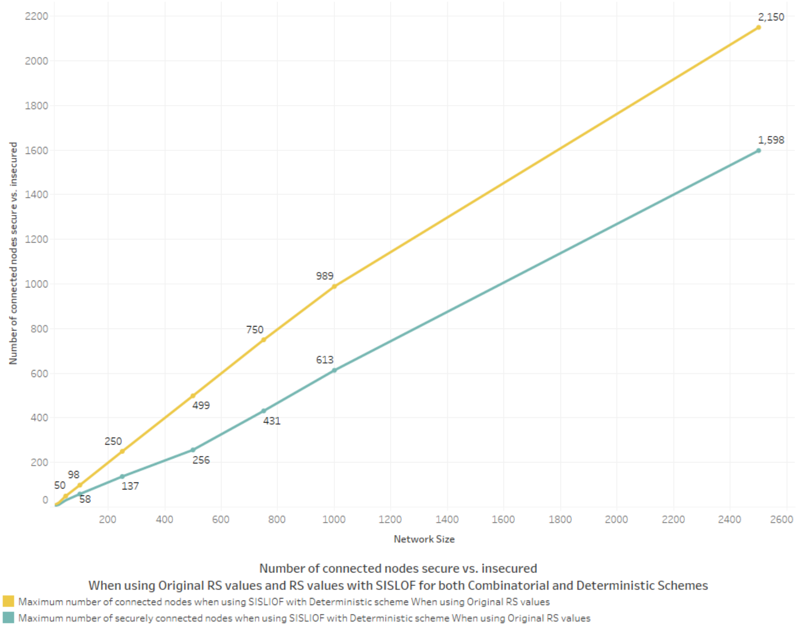


Figure 6.8: Number of DAGs in the DODAG (NNC) vs the number of secured when using *RS* computed in equation 3.5.1 with Deterministic scheme.

6.3.2 Network Connectivity With RSs Computed When Using Deterministic Scheme

In this section we evaluate the performance of the DAG in term of the number of nodes that share a key and the number of nodes that are securely connected. It is noticed that the DAG achieves full connectivity for all network sizes evaluated and when using various pools. The experiment result means that the ring size needed to achieve full connectivity when using SISLOF is smaller than the ring size values obtained in Chapter 5 for both ETX and OF0. In the next section we investigate the minimum ring size values needed to achieved full connectivity of the DAG.

6.3.3 Network Connectivity With IoT RSs

In the previous section we have identified that the ring sizes needed when using the Probabilistic scheme for SISLOF is smaller than the ring sizes computed in Chapter 5 . In this section we investigate the ring size needed to achieve full connectivity for each network size when using various pools by starting from the ring size values computed in Equation 3.5.1 as we have already identified in Section 6.3.1 that when using those ring size values the DAG contained a higher number of nodes securely connected however it did not reach full connectivity of the network.

We observe in Figure 6.9 that for all experiments for the different network and pool sizes that the ring sizes needed to achieve full connectivity is smaller than the ring sizes needed to achieve full connectivity for both ETX and OF0. We also note that the ring size values for all networks when using various pools is nearly half the ring sizes needed for ETX and OF0 and in smaller networks this increases to nearly three times smaller. Similarly to previous experiments the ring size decreases when using SISLOF in relation to OF0 is a smaller decrease than the decrease when comparing to ETX. This is related to the fact that OF0 will choose its preferred parent based on the parent rank only and does not take into consideration the quality of the link.

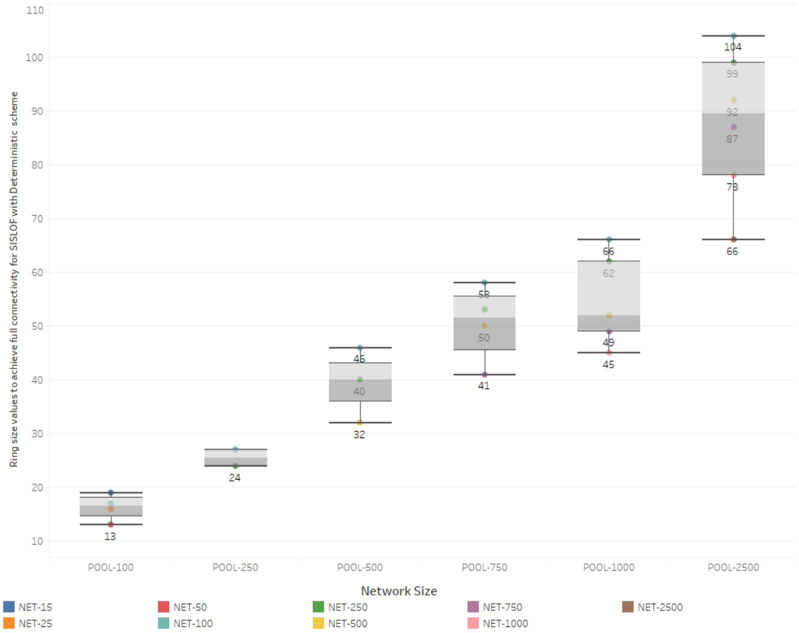
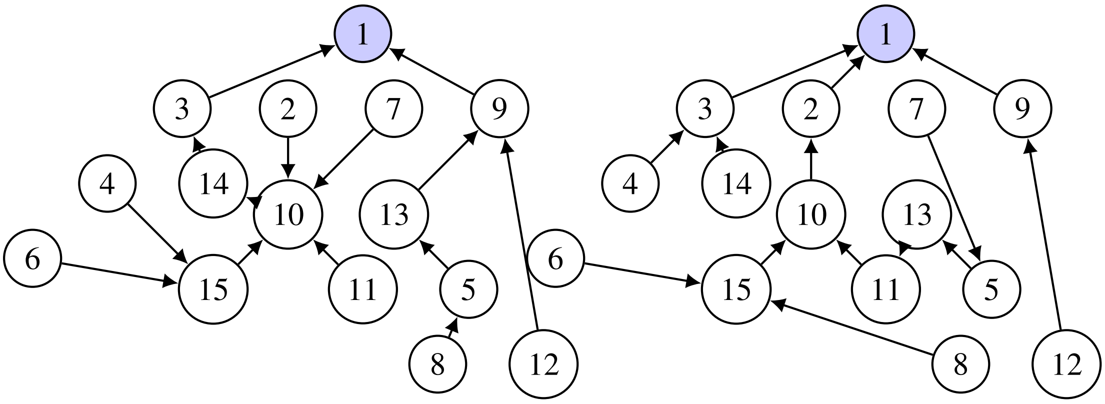


Figure 6.9: Ring size values to achieve full connectivity of network for Deterministic scheme. The ring size increases when the network size increases and the pool size decreases.

6.4 Fixed Network Experiment

In this section we evaluate the performance of the SISLOF objective function for both Probabilistic and Deterministic schemes in the fixed network experiment as presented in Section 3.7. Due to the nature of the SISLOF objective function, we observe that all nodes are securely connected for both Probabilistic and Deterministic schemes. We also note that when using SISLOF and the Probabilistic scheme, the DODAG shown in Figure 6.10a has a different topology from the topology of the network when using the Deterministic scheme shown in Figure 6.10b. Furthermore, the number of hops when using the Deterministic scheme is larger than the number of hops when using the Probabilistic scheme. This is due to the fact that the Deterministic scheme adds the FMAP variable and discard nodes that are considered not trusted. This reduces the number of nodes that can be considered a preferred parent and forces nodes to choose a node with lower rank.



(a) Secure DODAG using SISLOF with Probabilistic scheme

(b) Secure DODAG using SISLOF with Deterministic scheme

Figure 6.10: DODAG using SISLOF for both Probabilistic and Deterministic schemes showing full connectivity of the network.

6.5 Evaluation Of The Impact Of The RS In An IoT Environment

In this section, we evaluate the impact of the different ring sizes obtained in Tables 6.2 and 6.3 on the IoT DODAG formation and the individual nodes in the network.

6.5.1 RPL control Messages Number

In this section, we measure the number of control messages generated until all nodes join the network securely using the SISLOF objective function with the Probabilistic scheme used in Chapter 4 and the Deterministic scheme used in Chapter 5. The average control messages received for all nodes is a variable that will keep on changing as long as the simulation of the network runs. The average number of control messages for SISLOF for both schemes is shown in Figure 6.11. The number of control messages when using SISLOF with the Deterministic scheme reveals for all pool sizes a decrease in all control messages number in comparison with the Probabilistic scheme. This is due to the fact that in the Deterministic scheme each node will commit to each identity discovered in its neighbourhood and knows that it shares a key with it using the mutual authentication protocol FMAP. This reduces the number of control messages as nodes have more information and they dont need to predict as such whether they contain a shared key or not.

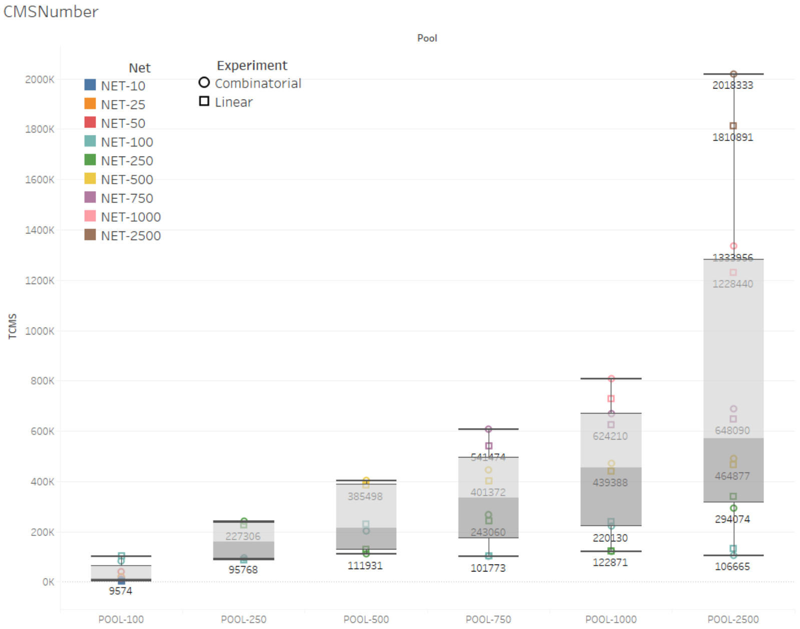


Figure 6.11: Number of RPL Control messages generated in 24 hours for different networks using Probabilistic scheme. The number of control messages increase when the network size increase for each pool size.

Similarly to previous experiments, we also note in Figures 6.12 and 6.13that the number of control messages is not directly related to the pool size but related indirectly to the ring size since the aim is to obtain a secure DODAG and thus nodes sharing a key is an important factor that will impact how many control messages each node will need in order to determine the best path and preferred parent.

As of the DSN networks, the number of packets is high simply because each node will need to send a packet to all other nodes in the network. As soon as the RPL routing table converges, the number of packets drops for each node. For experiments using SISLOF, the number of packets drops simply because each node is only sending to nodes that it shares an identifier with. This shows that the number of packets received per node after RPL converge gradually decreases when using SISLOF. This is shown in Figure 6.14. It is also essential to

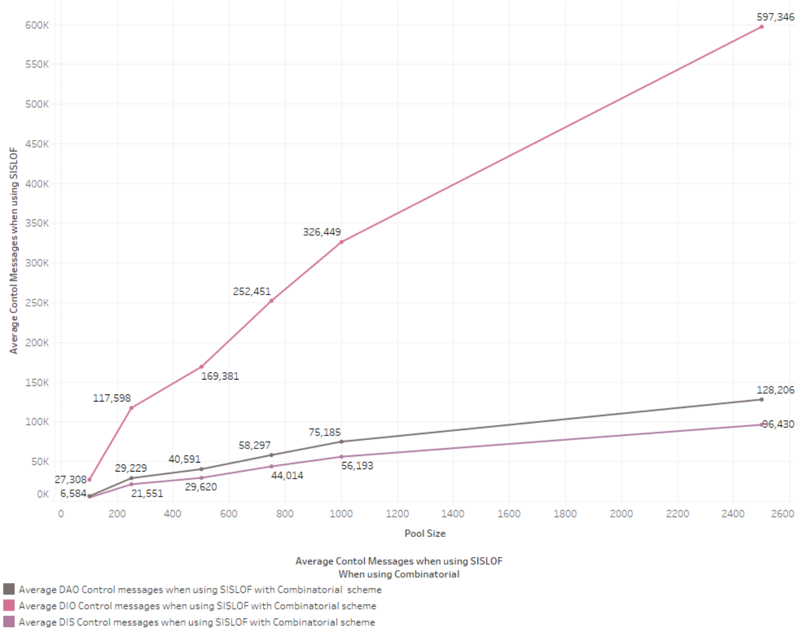


Figure 6.12: Number of RPL Control messages generated in 24 hours for different networks using Probabilistic scheme. The number of control messages increase when the pool size increases. The number of DIO messages is considerably larger than DAO and DIS.

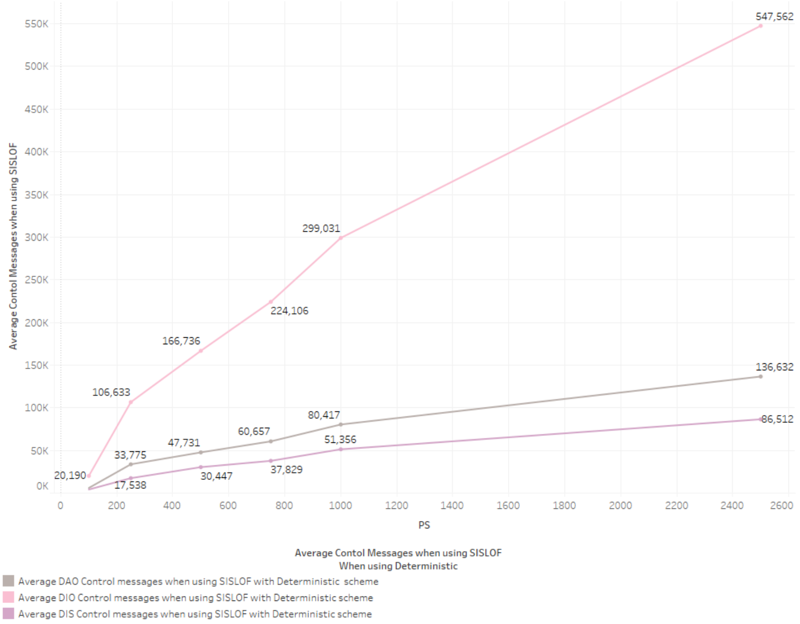


Figure 6.13: Number of RPL Control messages generated in 24 hours for different networks using Deterministic scheme. The number of control messages increase when the pool size increases. The number of DIO messages is considerably larger than DAO and DIS.

mention that the packet delivery radio for SISLOF is lower in all networks size as many RPL messages will be discarded since SISLOF objective function is used and therefore control messages coming from nodes that do not share a key with the recipient node will ignore it. The results in Figure 6.14 also reveals that there is no significant differences in packet delivery ratio between Probabilistic and Deterministic schemes except in the large network simulation of 2 500 due to the fact that in the Deterministic scheme it drops in addition to insecure messages FPMA messages for authentication with nodes that it shares a key with but are not in the RPL DAG instance.

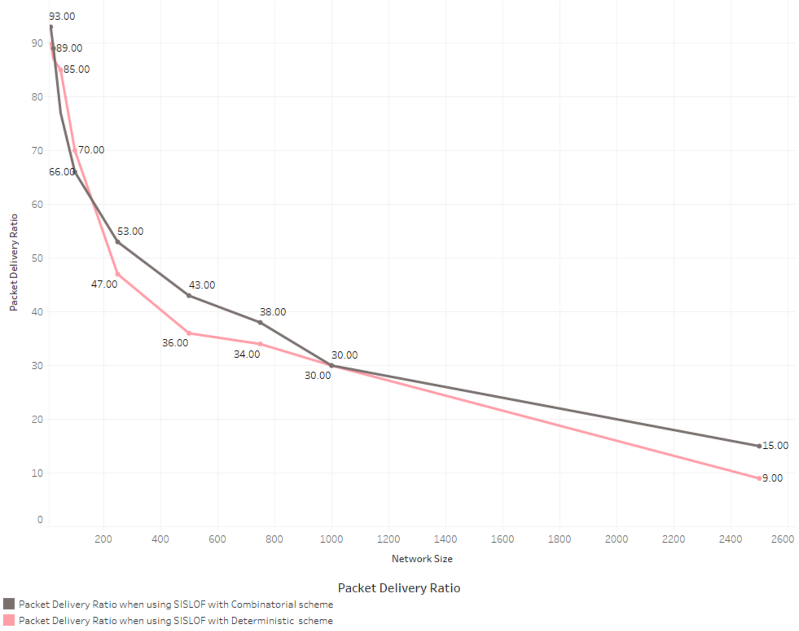


Figure 6.14: Packet Delivery Ratio of control messages for both Probabilistic and Deterministic schemes. The Probabilistic scheme outperforms the Deterministic scheme when using SISLOF for large networks.

6.5.2 Hops Count Average

In this section we compare the average hops count when using SISLOF with the Probabilistic scheme as a method to distribute the keys. We look at the initial hops count in the first five minutes in comparison with the hops count after the RPL converges. Since SISLOF selects the suitable path based on the number of hops to reach the root node we expect it to outperform OF0 and ETX. As before, we consider that an objective function underperform another when looking at the number of hops count the objective function that has a smaller hops count.

When running the experiment with Probabilistic scheme we observe in Figure 6.15 that the average hop count decreases when the pool size increases relatively to the network size. The decrease in the hops count decreases since the number of keys is increasing in the ring size because of the larger pools and the density of the nodes is also increasing because the simulation area is fixed and a larger number of nodes will naturally mean the nodes are closer to each other and each node when joining the DODAG will have more choices in term of which node to choose as a parent node.

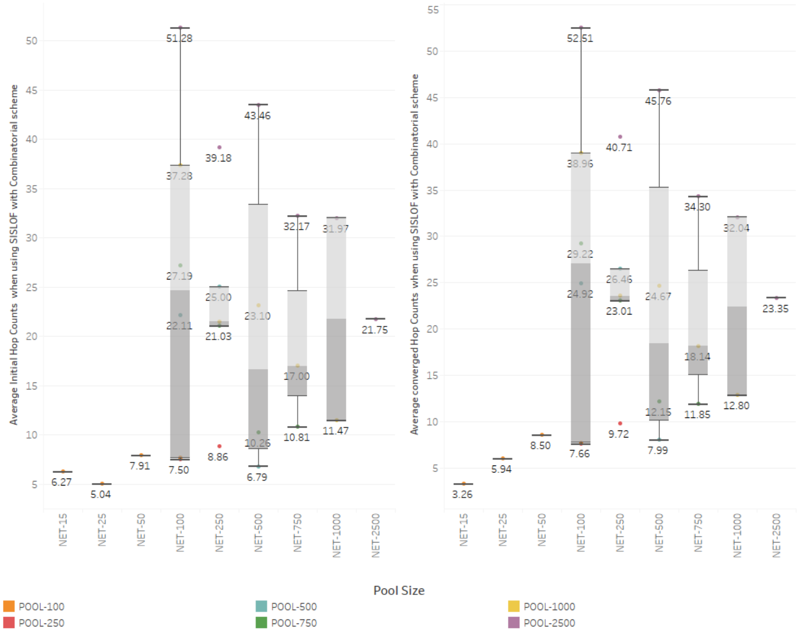


Figure 6.15: Initial and converged average hops count using SISLOF when using Probabilistic scheme. The hop count increases for all network sizes when the pool size increases.

Similarly to the average hop count when using the Probabilistic scheme, we note in Figure 6.16 that the average hop count decreases when using the Deterministic scheme and when the pool size increases relatively to the network size.

In Figure 6.17 the average hop count comparison for both the Probabilistic and Deterministic schemes does not reveal a significant difference in both the initial and the converged RPL DODAGs. This result provides a good fit to the expectation that the average hop count is related to how RPL DODAG is formed after the keys are distributed and the preferred parent is chosen.

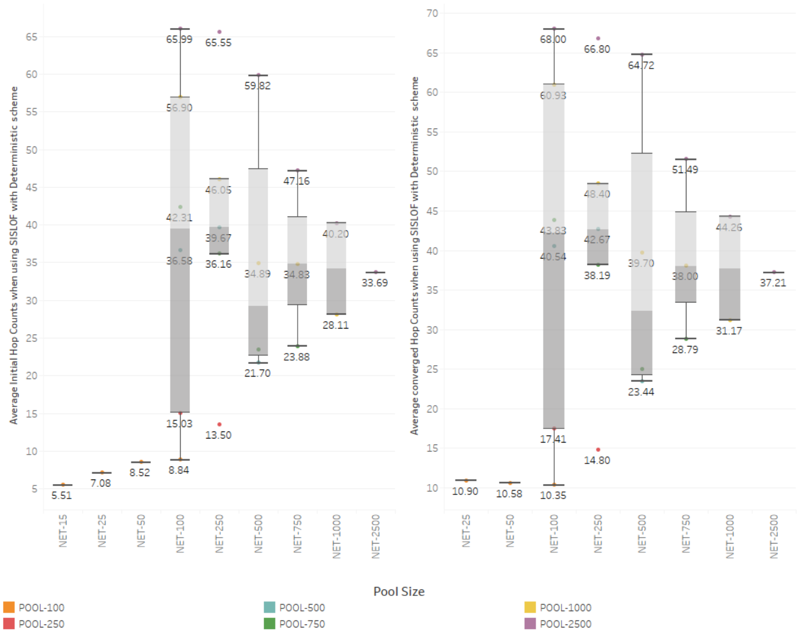


Figure 6.16: Initial and converged average hops count using SISLOF when using Deterministic scheme. The hop count increases for all network sizes when the pool size increases.

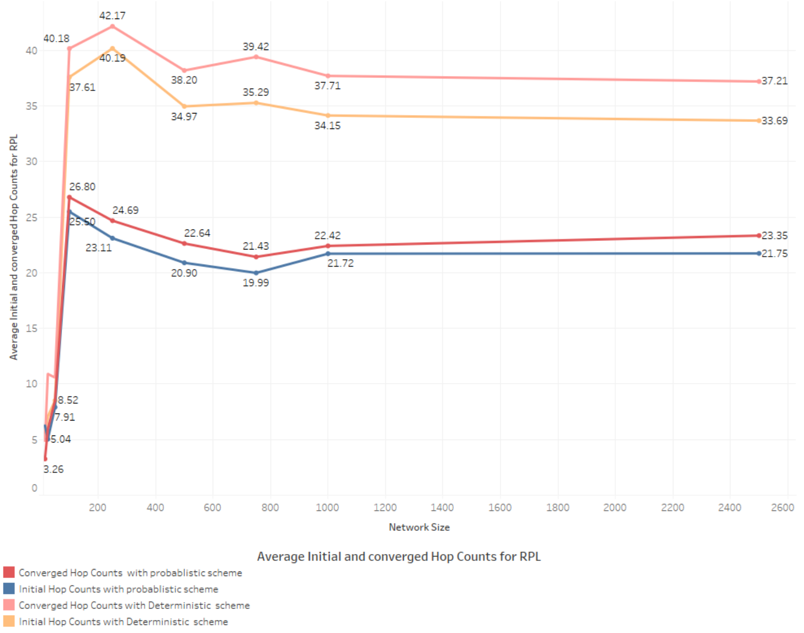


Figure 6.17: Initial and converged average hops count using SISLOF for both Probabilistic and Deterministic schemes. The hop count increases for all network sizes when the pool size increases.

6.5.3 Power Consumption

In this section we look at the average power consumption when using SISLOF. We measure power consumption similarly to the previous chapter using collected values from Powertrace and the Energest tool. We also look at the power consumption for the four components that consume the most in term of power consumption which are transmitting power (Tx), receiving power (Rx), the low power mode (LPM) state and the power consumption when the CPU is used. We note in Figures 6.19 and 6.18 that the power consumption distribution is proportional to what was forecasted. The biggest power consumption is in the receiver and transmitter for each device in comparison with the CPU and LPM power consumptions. The increase of the power consumption on all components in relation to the network size is also what was expected.

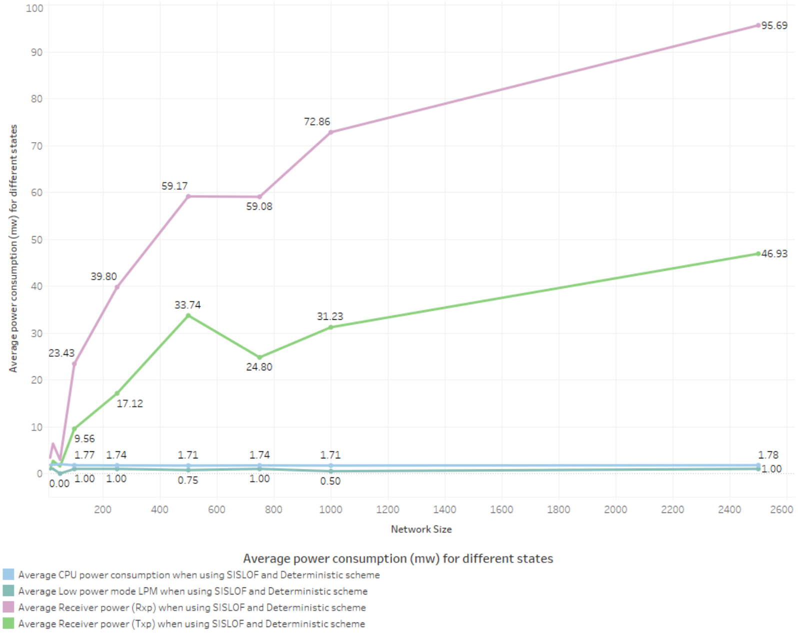


Figure 6.18: States power consumption when using Deterministic scheme. The states power consumption increases when the network size increases.

When looking at the total power consumption in relation to the pool size shown in Figures 6.20 and 6.21 we notice that the power consumption increases when the network size increases and is proportional to the increase of the pool size as well. This increase is related to the increase in the number of packets exchanged that is needed for each node in order to join the RPL DODAG.

Finally we observe in Figure 6.22 the average radio duty cycle in comparison with the average power consumption for all nodes. We note that there is a significant difference to

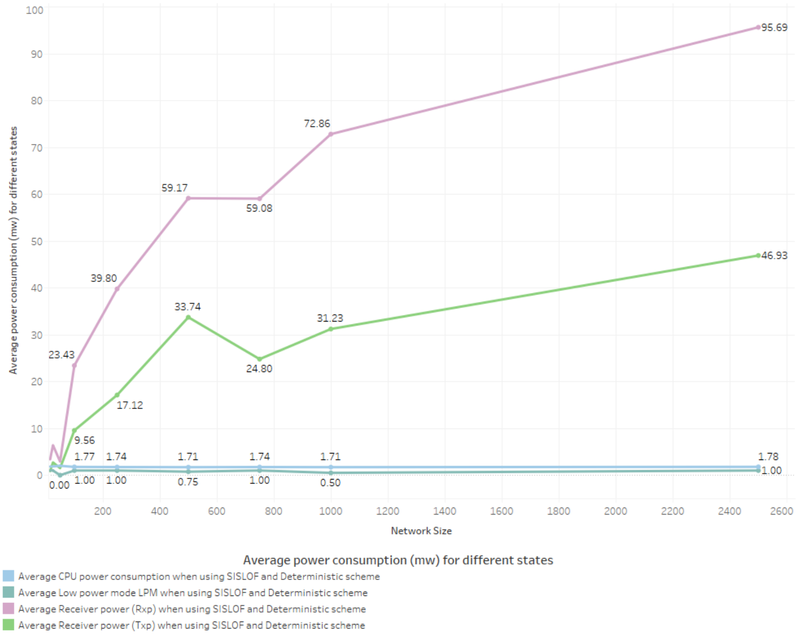


Figure 6.19: States power consumption when using Probabilistic scheme with SISLOF. The states power consumption increases when the network size increases.

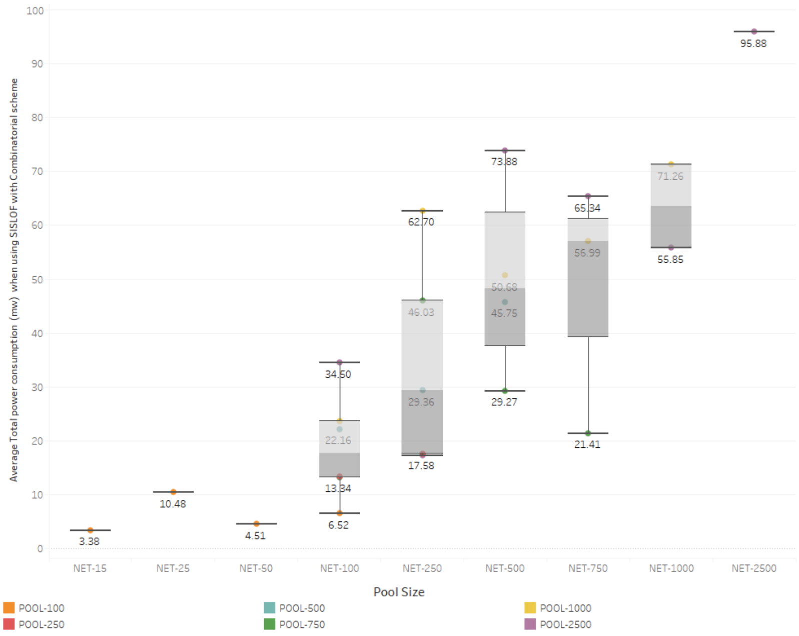


Figure 6.20: Average power consumption for Probabilistic scheme when using SISLOF. The power consumption increases when the network size increases for each pool.

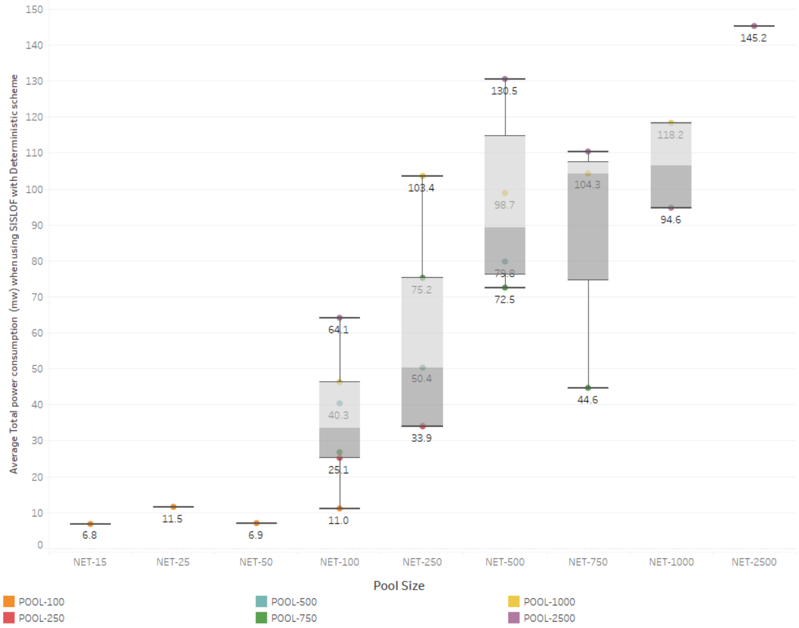


Figure 6.21: Average power consumption for Deterministic scheme when using SISLOF. The power consumption increases when the network size increases for each pool.

previous experiments simply because the power consumption increased significantly between the Deterministic key scheme and the Probabilistic key scheme.

6.5.4 CPU Usage

In this section we evaluate the CPU usage both when the RPL DODAG is forming and once it converges. As observed in Chapter 4 and 5, we observe in Figure 6.23 and Figure 6.24 that the CPU usage increases when the network size increases. This is expected since an increase in the number of nodes will force nodes to generate more RPL control messages in order to identify a preferred parent. We also identify that both the initial CPU usage and the converged one do not relate to the ring size.

We also note in Figure 6.25 that OF0 outperforms ETX in terms of CPU usage both for the average initial CPU usage and the average converged CPU usage. This is because in ETX objective function each node computes the suitable preferred parent by looking at several metrics for all nodes within its range it shares a key with and is trusted in the Deterministic key scheme. This adds more overhead on the CPU usage in comparison with OF0 that only rely on the link metric and the rank in the network to choose a suitable parent.

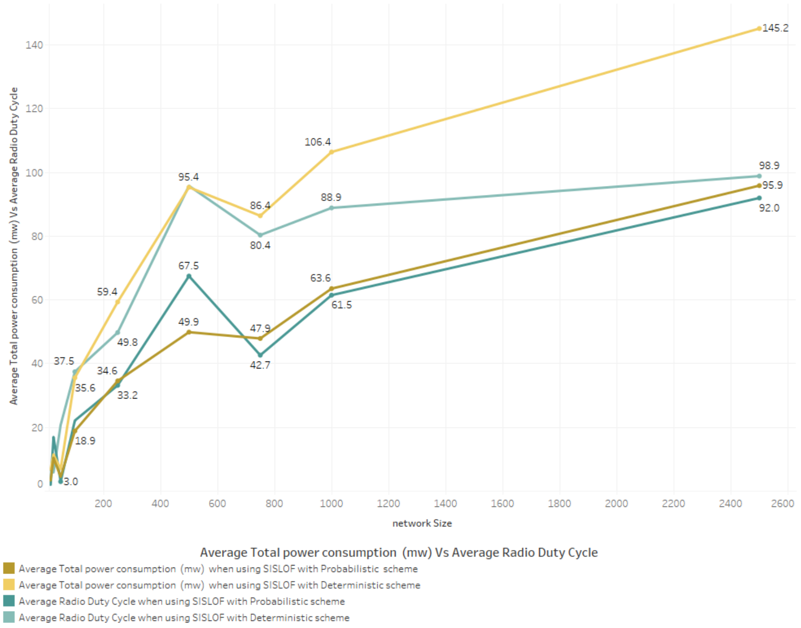


Figure 6.22: Average power consumption vs average radio duty cycle when using Probabilistic and Deterministic schemes. The average total power consumption for Probabilistic scheme outperforms the Deterministic scheme for various network sizes.

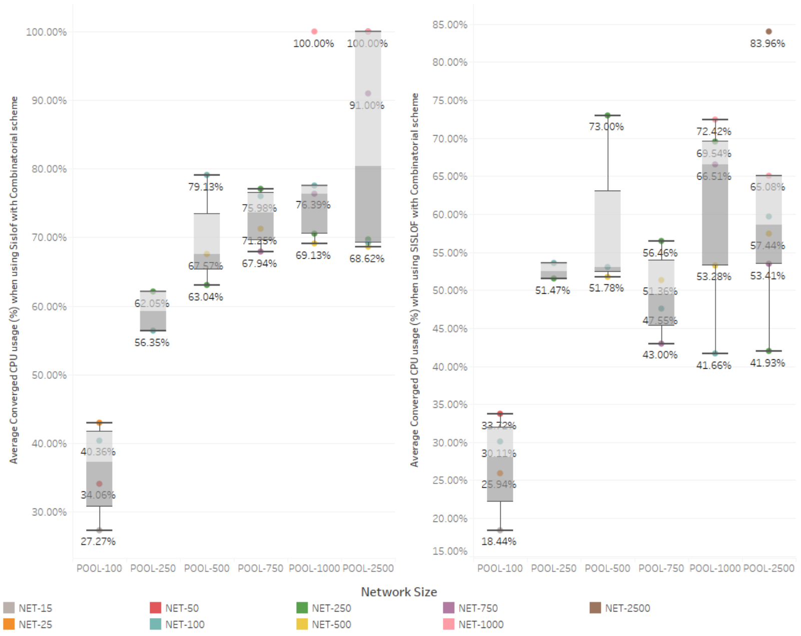


Figure 6.23: Initial CPU usage and converged CPU usage for RPL using Probabilistic scheme for different pools size. The CPU usage increases when the network size increases for each pool.

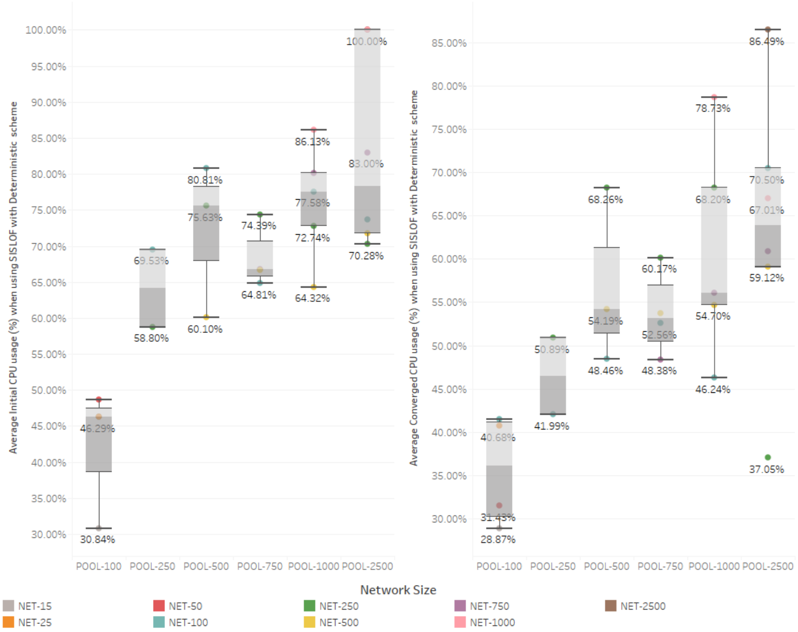


Figure 6.24: Initial CPU usage and converged CPU usage for RPL using Deterministic scheme for different pools size. The CPU usage increases when the network size increases for each pool.

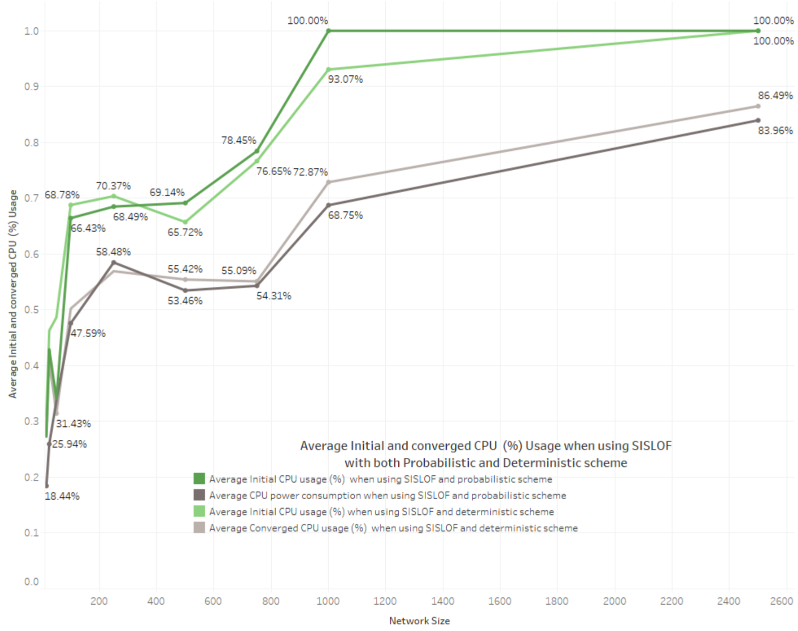


Figure 6.25: Initial CPU usage and converged CPU usage for RPL using both Probabilistic and Deterministic schemes. Deterministic scheme outperforms the Probabilistic scheme for the CPU usage.

6.5.5 Time to Converge

In this section we investigate the time it takes for the RPL DODAG to converge when using SISLOF using both the Probabilistic scheme as in Chapter 4 and the Deterministic scheme as in Chapter 5. We follow the same procedure to obtain the time to converge variables as we did in the previous chapters and we run all experiments for 24 hours to ensure that no significant changes in the RPL occur and to evaluate the time it takes for the RPL DODAG.

We investigate the time it takes for the DODAG to converge when using both ETX and OF0. We follow the same procedure to obtain the time to converge variables as we did in the previous chapter and we run all experiments for 24 hours to ensure that no significant changes in the RPL occur. We observe in Figure 6.27 and Figure 6.26 that ETX and OF0 takes nearly the same time to converge although the number of hops in OF0 is greater. This is due to the fact that OF0 takes more time to due to computation of the suitable link while ETX takes time to determine suitable node by comparing the node metrics of all neighbours of one specific node.

We note in figure 6.26 that the the Time To Converge (TTC) decreases when the pool size increase for each network size. This is due to the fact that more nodes share a key if the pool size is larger and this result is consistent with what we obtained in previous experiments in Chapter 4 and 5.

We also observe in Figure 6.27 that the the Time To Converge TTC decreases when the pool size increase for each network size. This is due to the fact that more nodes share a key if the pool size is larger and this result is consistent with what we obtained in previous experiments in Chapters 4 and 5.

We observe in Figure 6.28 that the RPL DODAG formation for the Deterministic scheme outperforms the Probabilistic scheme. This is because the FMAP authentication in phase one in identifying the trusted neighbours and results in the decrease in the number of nodes that will compete to be a preferred parent.

6.5.6 Latency

In this section we evaluate the latency when using SISLOF for both the Probabilistic and Deterministic schemes. We note in Figure 6.29 that the Probabilistic scheme outperforms Deterministic. This is due to the mutual authentication FMAP that determines trusted neighbours based on a voting process and therefore discards nodes that are considered not trusted even if they share a key. This results in an increase in the average number of hops

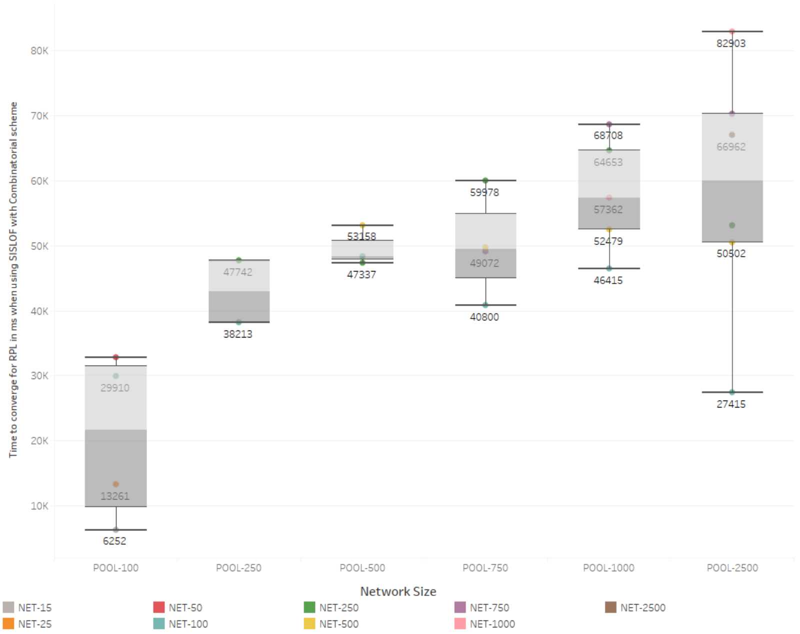


Figure 6.26: RPL DODAG time to converge when using Probabilistic scheme for different pools size. The time to converge for SISLOF when using Probabilistic increases when the network size increase and the pool size increase.

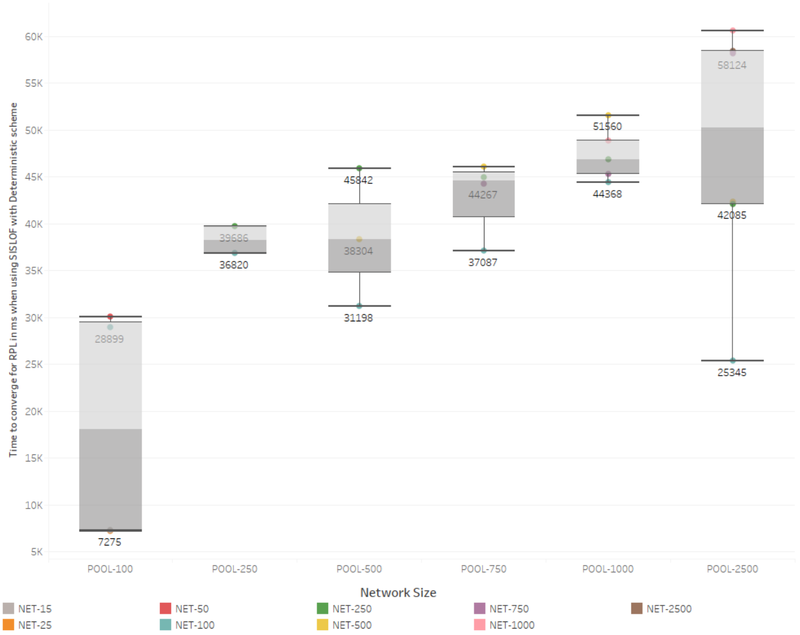


Figure 6.27: RPL DODAG time to converge when using Deterministic scheme for different pools size. The time to converge for SISLOF when using Probabilistic increases when the network size increase and the pool size increase.

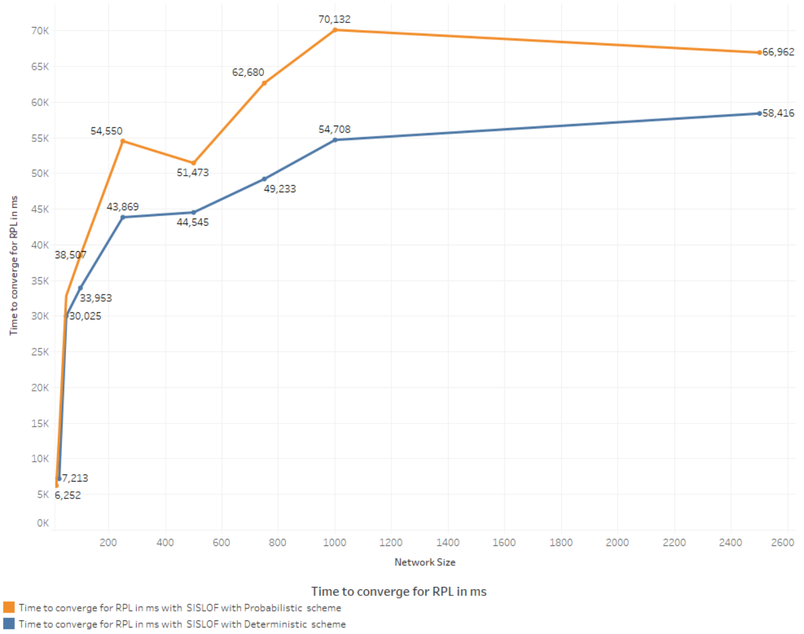


Figure 6.28: RPL DODAG average time to converge for both Probabilistic scheme and Deterministic scheme in different networks. SISLOF using Deterministic scheme outperforms the Probabilistic scheme there are less neighbours that can be potential preferred parents.

count in comparison with the Probabilistic scheme and results in an increase in the latency values.

We also note that the latency values for all experiments when using SISLOF is larger than when using ETX and OF0 for both Probabilistic and Deterministic schemes as the number of nodes increased when using SISLOF.

6.5.7 Number Of Neighbours

In this section we investigate the number of neighbours that share a key in the network. We note in Figure 6.30 that the difference in the number of neighbours (NNE) between Probabilistic and Deterministic schemes is not very large. This is because of the voting process in the FMAP authentication phase where some of the neighbours are discarded even if they share a key because of the lack of trust and for this a mutual authentication cannot be agreed.

We also note that the number of neighbours that share a key for both the Probabilistic and Deterministic scheme when using SISLOF decreased significantly in comparison with the number of neighbours observed in Chapters 4 and 5.

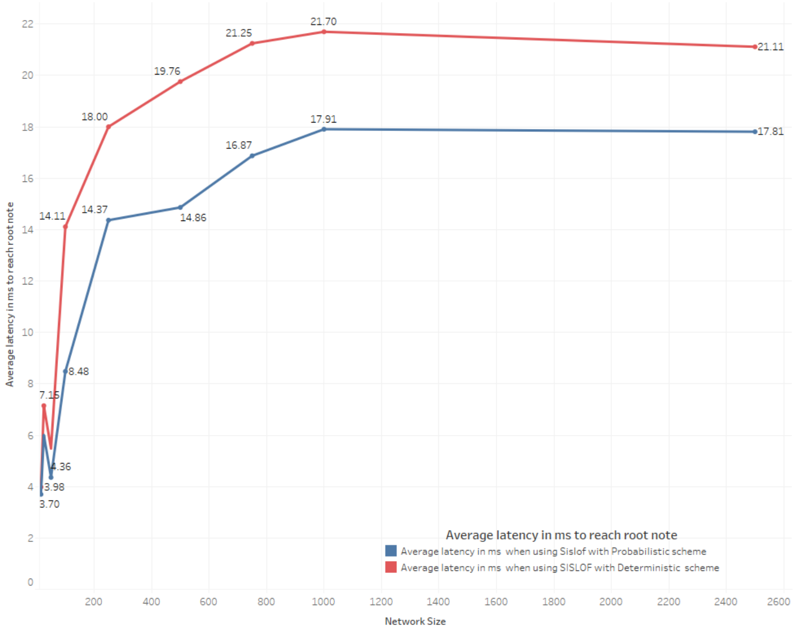


Figure 6.29: Average latency in ms when using Probabilistic and Deterministic schemes. Deterministic scheme outperforms the Probabilistic scheme since there are less neighbours which results in less number of hops and less time for packets to reach the root node.

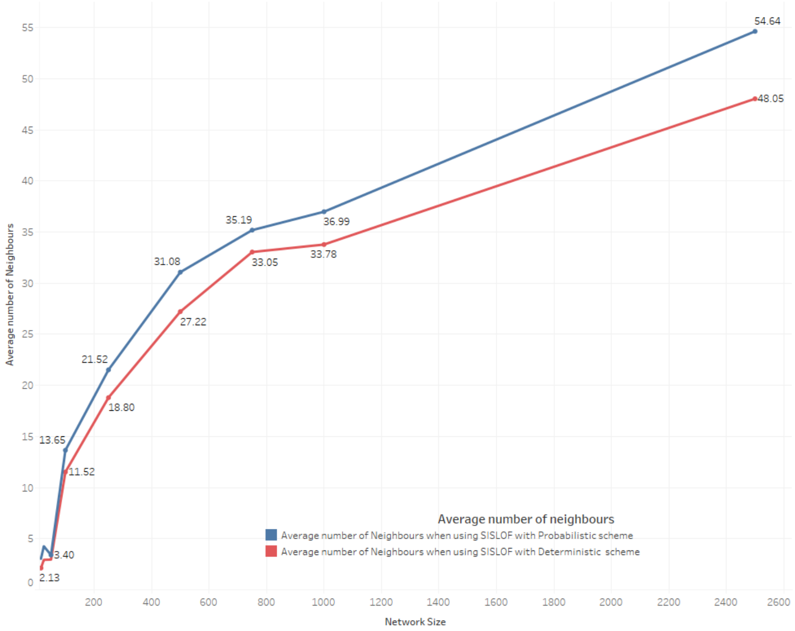


Figure 6.30: Average number of Neighbours when using Probabilistic and Deterministic schemes. Probabilistic scheme outperforms Deterministic scheme when using SISLOF since the Deterministic scheme discards some neighbours because of the voting process FMAP authentication algorithm.

6.6 Summary

In this chapter and in [3] a new objective function was introduced and its performance was evaluated in the context of both the Probabilistic and Deterministic scheme approaches.

We have identified that the rings sizes needed to achieve full connectivity when using SISLOF is smaller than the rings sizes for both ETX and OF0. We have also identified that the number of RPL control messages are smaller for ETX and OF0 than the ones for the SISLOF. The link metric for SISLOF underperforms ETX and OF0 such as latency, time to converge and the the number of hops. However SISLOF outperform ETX and OF0 in all node metrics such as the CPU consumption and the power consumption.

The voting process in the FMAP authentication phase for the Deterministic scheme has a clear impact on the performance of SISLOF when using it. The impact is the result of a decrease in the number of trusted nodes that are allowed to become potential parents or neighbours in the RPL DODAG. Since the number of neighbours decreased, the average number of hops increases as less neighbours are available. This results in an increase in latency as there are more hops in the RPL DODAG.

Chapter 7

Hardware experiment

In this chapter we evaluate the performance of the both the Probabilistic scheme and Deterministic scheme when using ETX, OF0 and SISLOF objective functions in a small environment of 15 Zolertia devices as discussed in Section 3.3. We evaluate using the Zolertia devices introduced in Figure 3.1a the performance of a small network environment when using the ring sizes obtained in both the Probabilistic scheme Chapter 4 and the Deterministic scheme in Chapter 5 and in the context of the different objective functions compared OF0, ETX and SISLOF.

Similarly to the previous chapters, we then evaluate the impact of the different ring sizes on the performance of the Zolertia device. We finally compare the ring sizes and the impact of those ring size when experimenting using Zolertia or when simulating the same number of devices in the simulated environment.

A summary of the objectives of this chapter are listed below.

1. Identify how the Probabilistic scheme performs with different ring sizes values for ETX, OF0 and SISLOF objective functions in Section 7.1 .
2. Evaluate the impact of the obtained ring sizes when using both Probabilistic and Deterministic schemes in Section 7.2 .
3. Evaluate and compare the results obtained for both a simulated environment and a physical environment with the same parameters in Section 7.2 .

7.1 Network connectivity with RSs in an IoT environment

In this section we evaluate the rings size for the physical environment in comparison with a simulated environment. We compare in the sections below the performance of the objective functions when using the Probabilistic scheme in Section 7.1.1 and the Deterministic scheme in section 7.1.2.

7.1.1 Rings size When Using Probabilistic Scheme

We compare in this section the rings size obtained in Chapters 4 and 6 for ETX , OF0 and SISLOF in the simulated environment with the physical environment introduced in this chapter when using the Probabilistic scheme. We note in Figure 7.1 that the rings size for the physical environment are smaller than the rings size for the simulated environment. The fundamental assumption for the difference is based on the fact that the distance between nodes in a simulated environment is more random than the physical environment and the distance between nodes is smaller for the physical environment than the simulated environment.

Similarly to results obtained in the simulated environment in previous chapters, we observe that the performance of the objective functions for the physical environment is consistent with the previous results. OF0 outperforms ETX since the link metrics are not a factor in deciding the preferred parent. SISLOF objective function outperforms both ETX and OF0 as well since the main factor that applies to form the DODAG is whether the connected nodes share a key or not.

7.1.2 Rings Size When Using Deterministic Scheme

In this section we investigate the performance of the objective functions in a physical environment when using the Deterministic scheme and we compare the results to the ones obtained in Chapters 5 and 6. We note in Figure 7.2 that the rings size for the physical environment are smaller than the ones obtained in the physical environment. This result is consistent with what we observed in the previous section.

7.1.3 Ring Size Evaluation

We show in Table 7.1 the series of experiments that resulted in computing the rings size needed to obtain a full network connectivity when running the experiments in a physical environment. We notice in Figure 7.3 that the Deterministic scheme outperforms the Probabilistic scheme for all objective functions. This is consistent with the rings size obtained for the simulated environment.

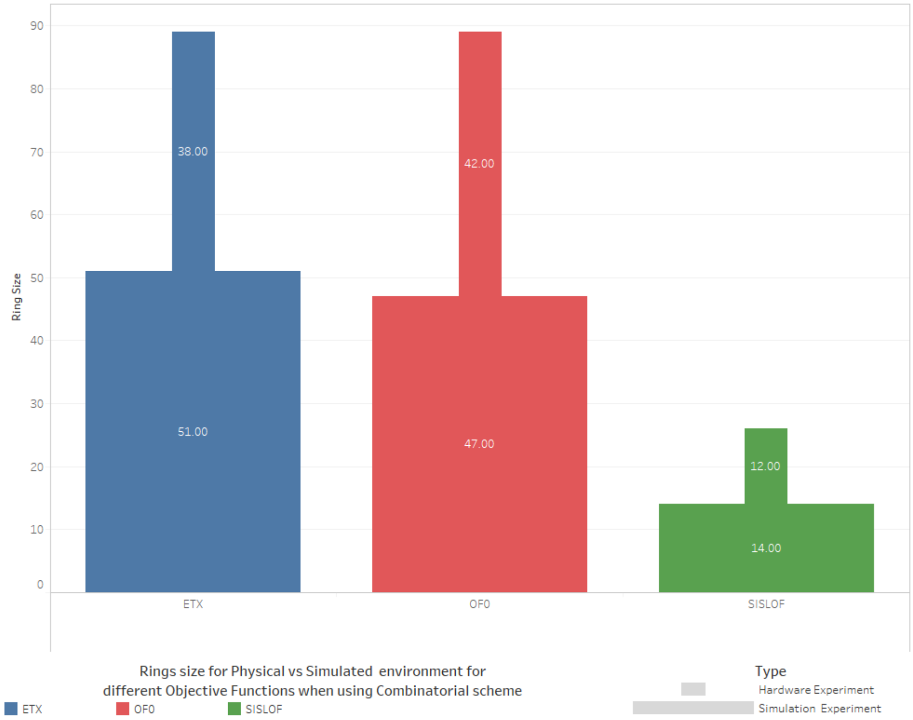


Figure 7.1: Rings size comparison when using Probabilistic scheme with ETX, OF0 and SISLOF objective functions for hardware and software environments. For all experiments, a smaller ring size is needed to achieve full connectivity in the hardware environment experiment in comparison with the simulated environment experiment.

Experiment		Probabilistic			Deterministic		
		RS	PSK	RS	RS	PSk	RS
Simulation	ETX	8	20%	51	8%	15	27
	OF0	8	28%	47	8	8%	17
	SISLOF	8	33%	14	8	33%	19
Hardware	ETX	8	33%	38	8	23%	17
	OF0	8	15%	42	8	16%	19
	SISLOF	8	73%	12	8	93%	9

Table 7.1: Ring size values for ETX, OF0 and SISLOF to achieve full connectivity with both Probabilistic and Deterministic and when using ring size values for DSN for Simulation and hardware experiments. The percentage of shared keys when using the ring size value of DSN are shown and no full connectivity is achieved.

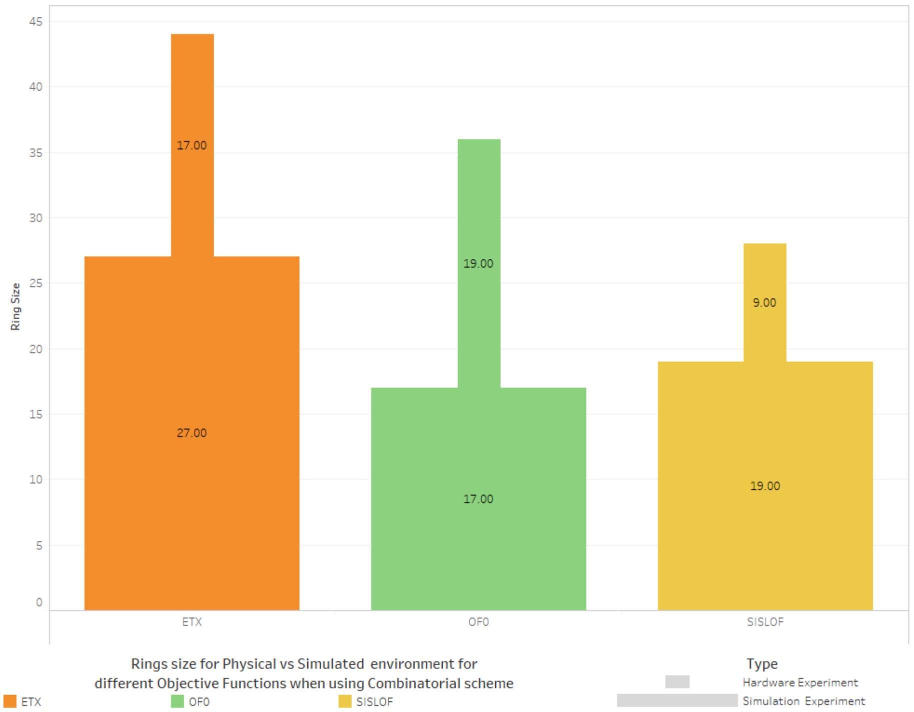


Figure 7.2: Rings size comparison when using Deterministic scheme with ETX, OF0 and SISLOF objective functions for hardware and software environments. For all experiments, a smaller ring size is needed to achieve full connectivity in the hardware environment experiment in comparison with the simulated environment experiment.

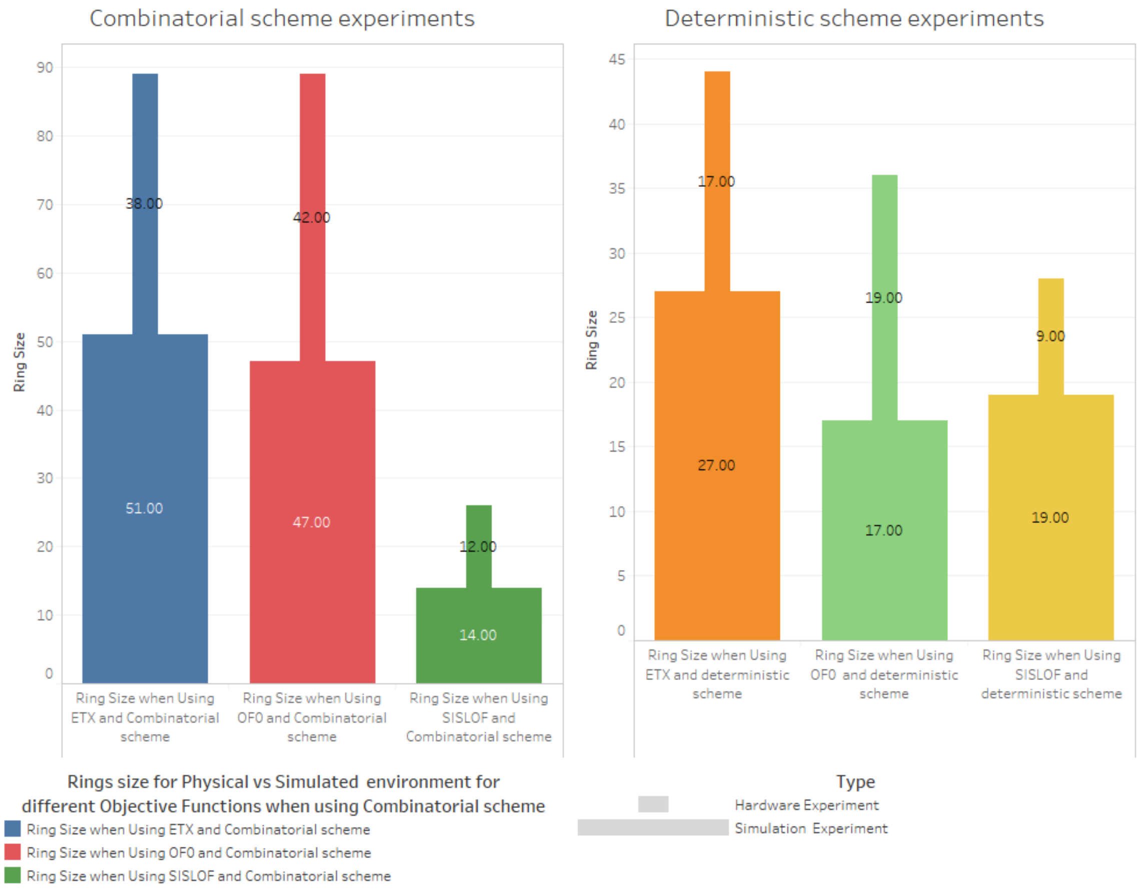


Figure 7.3: Ring size comparison for Probabilistic and Deterministic schemes when running the experiments in the simulated environment vs the hardware experiment. The ring sizes in the hardware experiment environment are smaller than the simulated environment for all objective functions.

7.2 Evaluation Of The Impact Of Increasing The RS In An IoT Environment

7.2.1 RPL control Messages Number

The main focus of the experiment in this section was to compute the number of RPL control messages when running the experiment in a physical hardware environment for 15 Zolertia nodes and we compare the results with the same number of nodes in a simulated environment.The experiment setup is generic as we follow the same experiments carried out with the ring sizes computed in Equation 3.5.1 for DSN networks out while using either ETX, OF0 and SISLOF with Probabilistic and Deterministic schemes and computed in Chapters 4, 5 and 6 and shown in Table 7.1 .

The control message numbers in Figure 7.4 confirms that it is not directly related to the pool size but related indirectly to the ring size since the aim is to obtain a secure DODAG. The control messages number in the hardware simulation also reveals that the DAO messages is significantly larger than the number of DIO and DIS messages. These values are confined to the fact the SISLOF will only accept nodes that it shares a key with before forming the routing table as opposed to OF0 and ETX where the large ring size values result in more neighbouring nodes sharing a key.

We undertake in Figure 7.5 the empirical analysis using data collected in previous simulated experiments in Chapters 4, 5 and 6. We note that the number of DAO control messages is significantly lower than the number of control messages for the same network size and pool size when running a physical experiment. This is due to the fact that the distance between nodes in a simulation is larger than the distance in the physical environment where most of the nodes were within reach from each other and thus were able to communicate by sending DAO messages to establish a route only with nodes that it shares key with and can establish a secure communication with. We an also observe that the number of control messages when using SISLOF is higher in both the practical and the simulation experiments in comparison with ETX and OF0 since SISLOF only exchange control messages with nodes that it shares key with in comparison with ETX and OF0 that exchange messages to identify if a shared key exists.

In Figure 7.6 we note that the total number of secured control messages transmitted for SISLOF in both practical and simulation schemes is higher than the total number of secured control messages for ETX and OF0. This is the result of having the DAG link only established if the nodes share a key and since the ring sizes for SISLOF is lower it

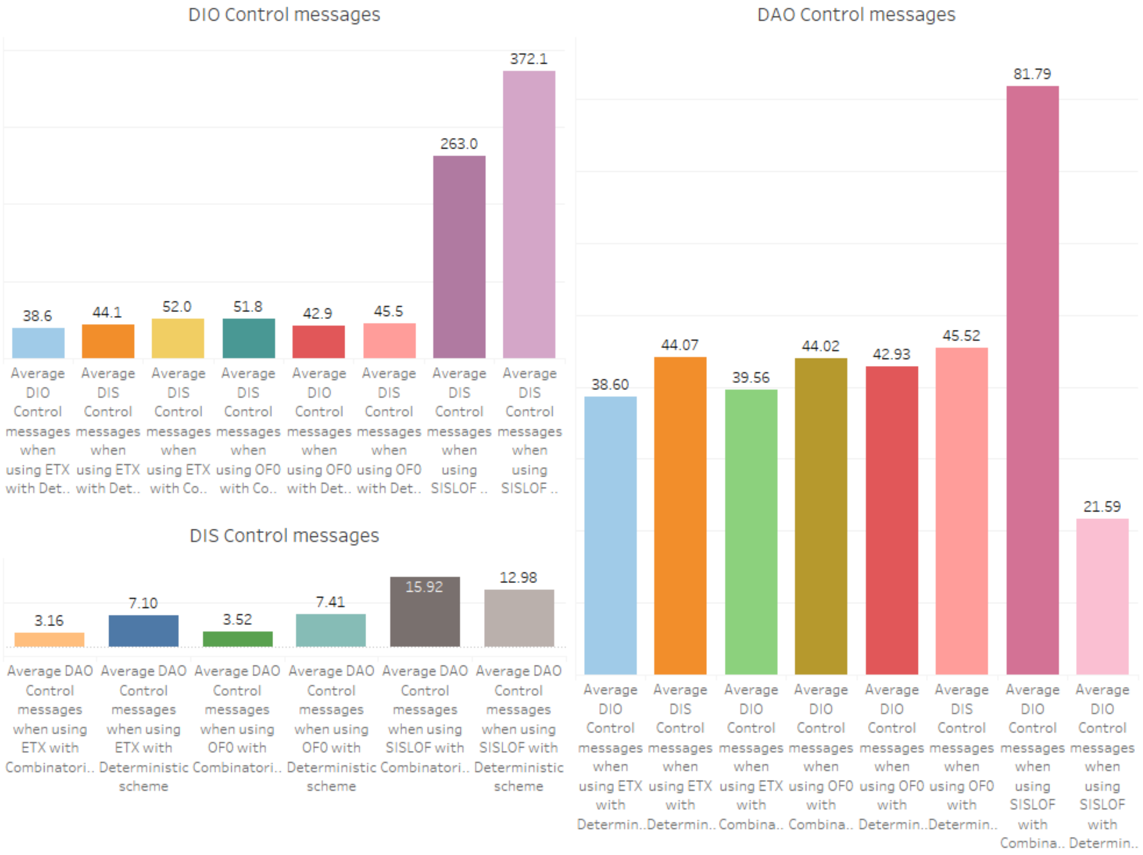


Figure 7.4: Number of control messages DAO, DIO, DIS for ETX, OF0 and SISLOF when using both Probabilistic scheme and Deterministic scheme for hardware experiment.

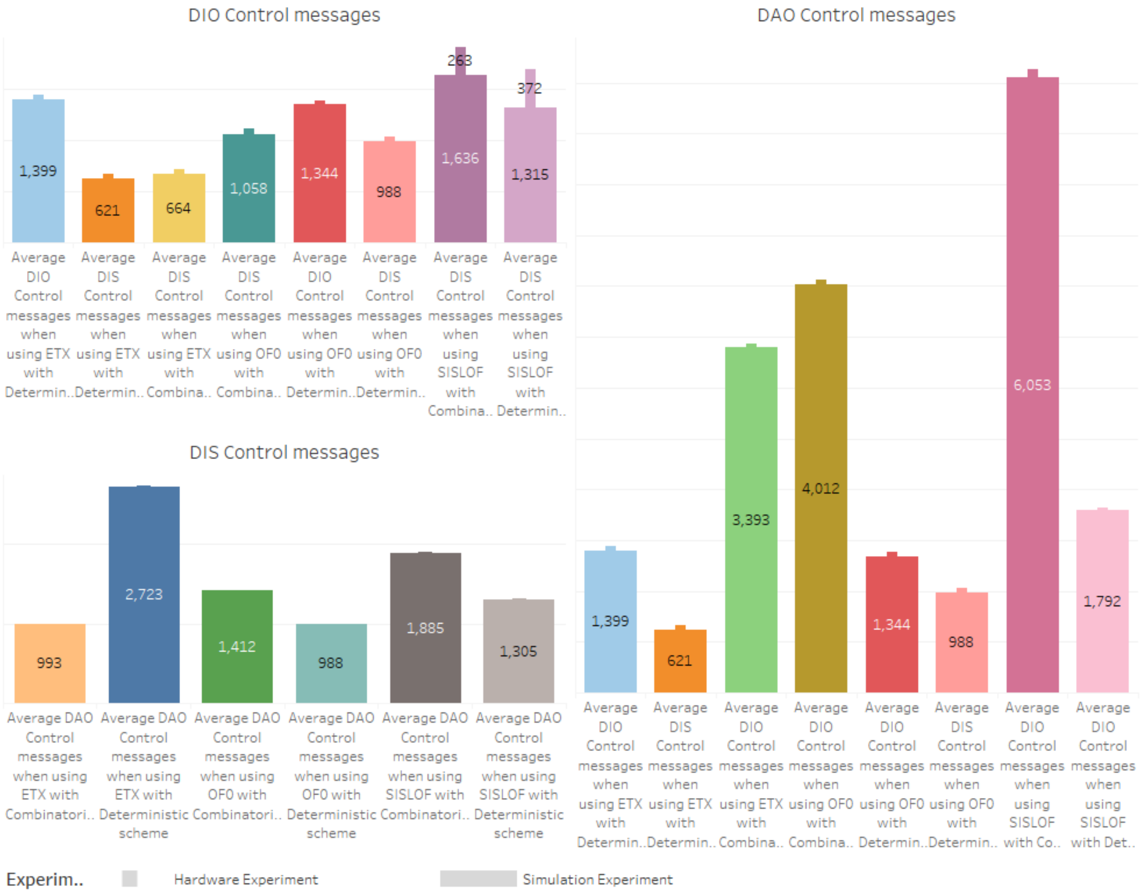


Figure 7.5: Number of control messages DAO, DIO, DIS for ETX, OF0 and SISLOF when using both Probabilistic scheme and Deterministic scheme for hardware and simulation experiments. The number of control messages when running hardware are lower than the number of control messages in the simulated environment for all experiments.

means more control messages to be sent for nodes to identify neighbouring nodes that they share a key with. We also note that the total number of secured control messages in the simulation experiments is higher for all objective functions when using both the Probabilistic and Deterministic schemes. This is because the physical environment where the network runs is smaller than the simulation experiment environment and the change of a node finding neighbours with shared keys without exchanging too many control messages.

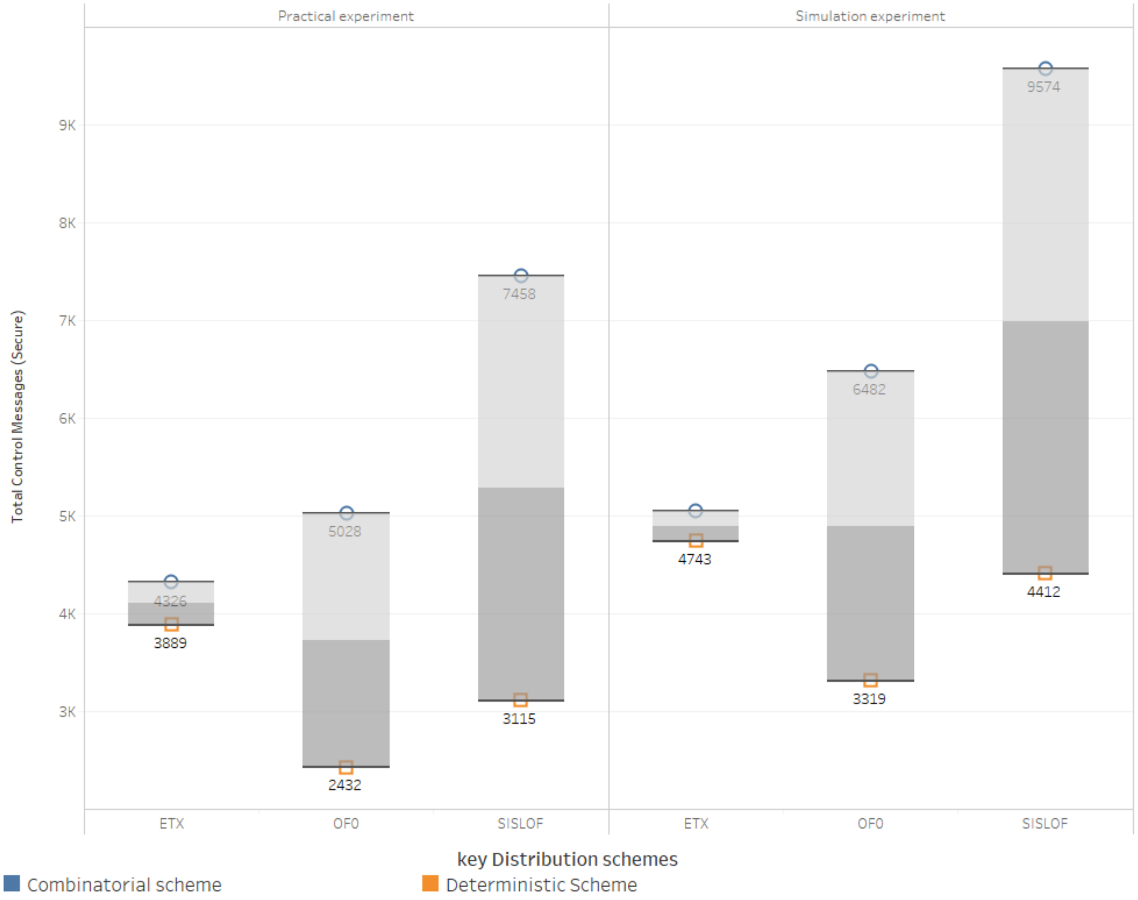


Figure 7.6: Total secured control messages transmitted for ETX, OF0 and SISLOF when using both Probabilistic scheme and Deterministic scheme for hardware and simulation experiments. The Deterministic scheme outperforms the Probabilistic scheme for both the physical and simulated environments and for all objective functions.

In Figure 7.7 we compare the packet delivery ratio for physical experiment with the simulation experiment packet delivery ratio data obtained from experiments in Chapters 4 and 5. We observe that the packet delivery ratio for the physical experiment is lower than the simulation experiment. This trend suggest that the impact of the relatively high number of control messages shown in Figure 7.6 result in control messages being dropped. This does not happen or happens less in simulation environment and although the number of control messages is higher is because the simulation environment is larger and nodes are more spaced with less neighbours for each node. This result in less packets being dropped or discarded by the receiving node.

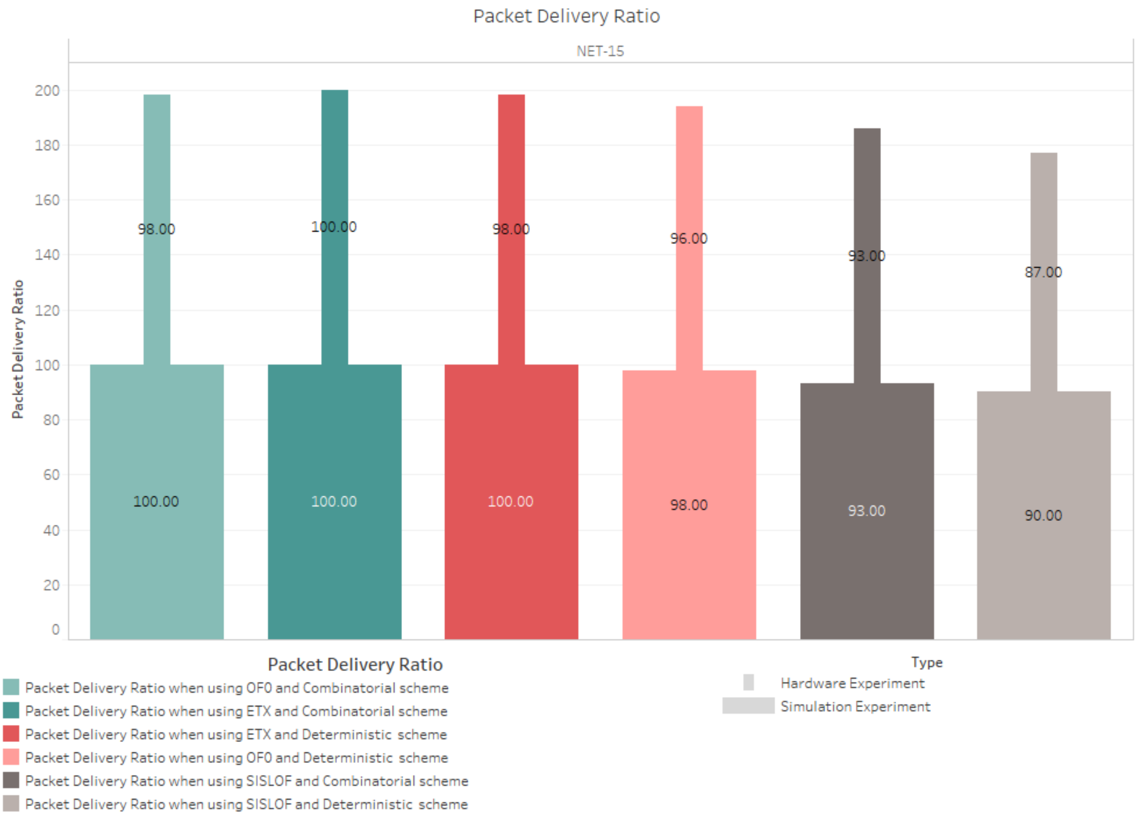


Figure 7.7: Packet Delivery Ratio of control messages for both Probabilistic and Deterministic schemes using ETX, OF0 and SISLOF for hardware and simulation experiments. SISLOF objective function outperforms both ETX and OF0 when using both Deterministic and Probabilistic scheme however the difference is clearer in the hardware environment.

7.2.2 Hops Count Average

In this section we compare the average hops count when running the experiments in the physical environment and the simulation environment for ETX, OF0 and SISLOF when using the Probabilistic scheme and Deterministic Scheme. We observe in Figure 7.8 that the Initial hops count is higher than the converged hops count for all experiments and is consistent when running this in the physical or simulated environment. We also observe that the hops count for all experiment in the physical experiment is lower than the simulated ones. We conclude that the the reason for the decrease in the hops count in the physical environment is due to the fact that nodes in the physical environment are closer to each other due to the nature of the environment they are running in and therefore more than one neighbouring node can share a key for each node. This results in SISLOF choosing a preferred parent that has a higher rank and therefore reducing the number of hops needed to reach the root node.

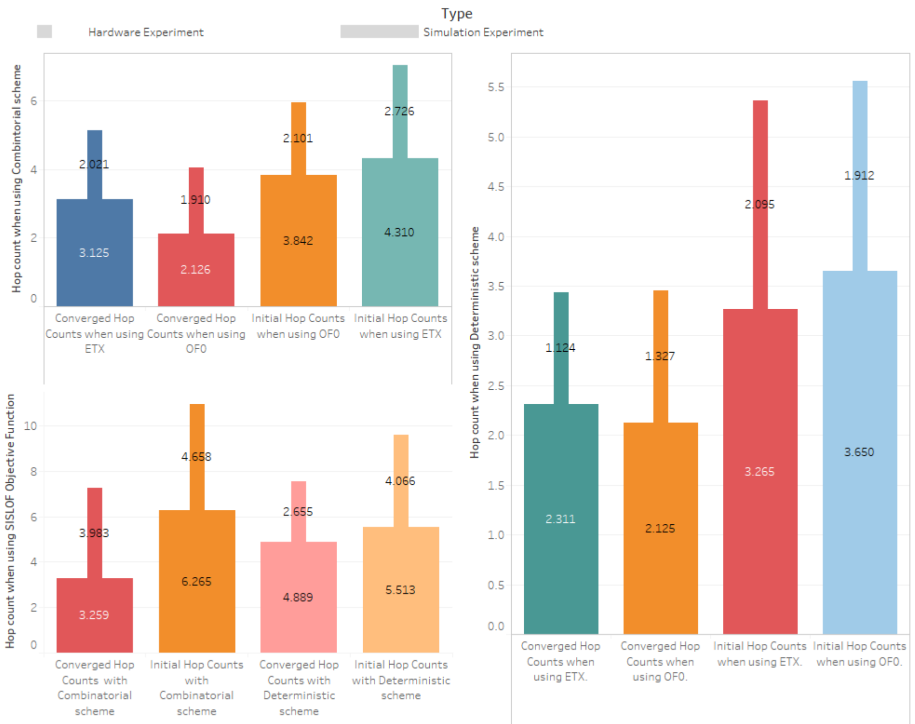


Figure 7.8: Initial and converged average hops count using RPL comparison for both the hardware and the simulation experiments. The average hop counts for all experiments is lower than the simulated environment ones when running the experiments in the physical environment.

7.2.3 Power Consumption

In this section we evaluate the power consumption metrics obtained in two different environments, the physical environment of 15 nodes and the simulated environment of the same number of nodes. We experiment with the three objective functions when using the Probabilistic and Deterministic scheme. We note for all objective functions in Figure 7.9 that in both the Probabilistic and Deterministic schemes the power consumption in the physical environment experiment is significantly larger than the simulation environment. This is due to two factors, the first is due to the close proximity of the devices in a physical environment due to the nature of the experiment. The second factor for which a hypotheses was developed, the physical device will naturally consume more power as it is subject to a more realistic condition and the power consumption in this case of all the compared variables of is higher in the physical environment. We note that this is the only variable that has this significant difference. We observe that the OF0 objective function outperforms the ETX objective function in both simulated and physical experiments which is a consistent result with what we have seen in all simulated experiments. We also note that the Deterministic scheme outperforms the Probabilistic scheme when using ETX and OF0 objective functions but underperforms when using SISLOF. This is due to the fact that when using the Deterministic scheme many nodes are discarded due to FMAP which results in lesser number of nodes that can be considered as preferred parent, however when using SISLOF the power consumption increases as the SISLOF objective function adds the secure link variable to the formation before nodes can join the RPL DODAG and therefore more RPL control messages are needed to establish the DODAG which increases the power consumption.

In Figure 7.10 we show the average power consumption and the radio duty cycle for both the the Probabilistic and Deterministic schemes for all objective functions when experimenting in the physical and simulated environments. In the previous section we identified that the total power consumption for SISLOF is higher than the total power consumption for ETX and OF0 when using the Probabilistic scheme and lower when using Deterministic scheme. We note in this figure that the average radio duty cycle when using SISLOF is relatively higher than the radio duty cycles for both ETX and OF0 however and opposite to the total power consumption, the average radio duty cycle when using SISLOF with Probabilistic scheme outperforms the average radio duty cycle when using SISLOF with the Deterministic scheme. The radio duty cycle increases when running in a hardware environment since the nodes are in closer proximity which forces nodes to wake up more regularly.

7.2.4 CPU Usage

In this section we evaluate the performance of the objective functions when using the Probabilistic and Deterministic schemes in term of the initial and converged CPU percentage and when running the experiments in both a physical and simulated environment for a network of 15 nodes and a pool size of 100 keys. We observe in Figure 7.11 that the trend is for the average percentage of CP usage to decrease in comparison with the Initial CPU usage. This is a predicted result as it matches with previous experiments done in Chapters 4, 5 and 6. We also note that the CPU usage for the nodes in the physical environment are

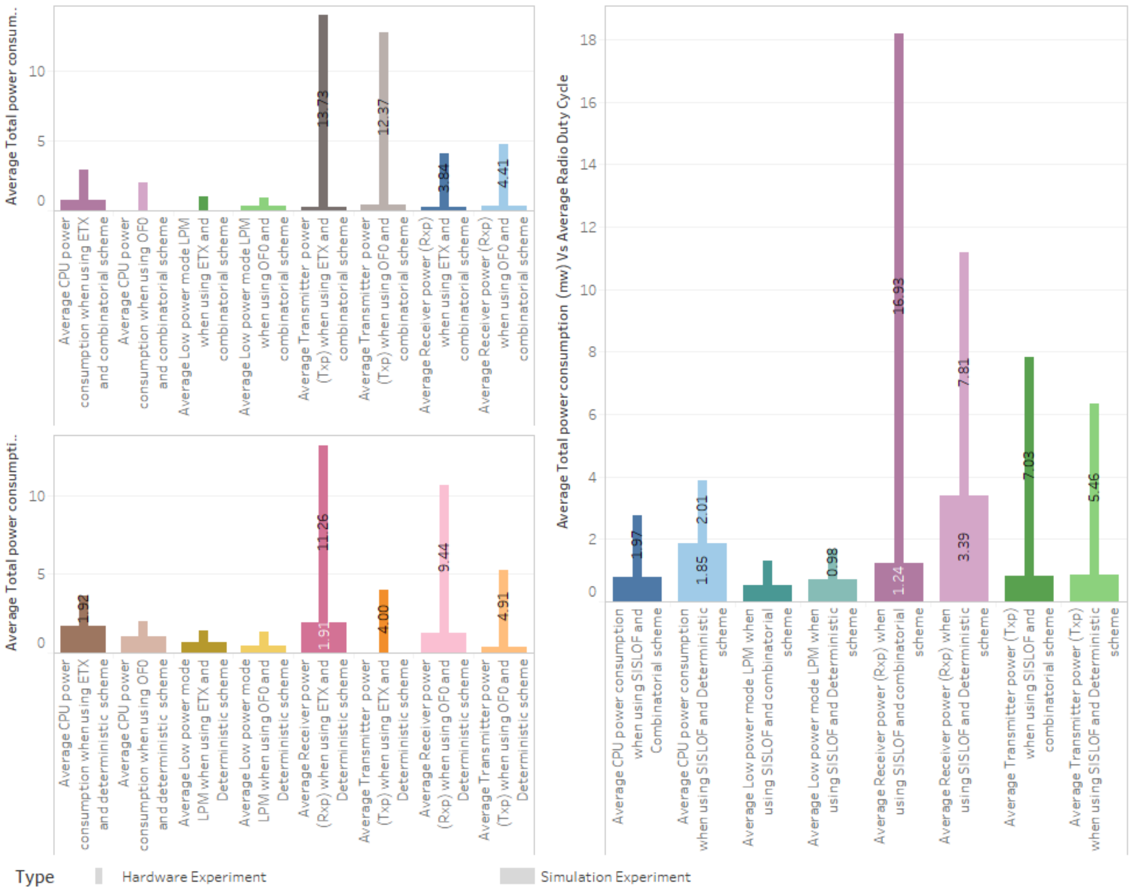


Figure 7.9: Average power consumption when using the Probabilistic and Deterministic schemes for all Objective functions. Power consumption for all power states are higher for the physical environment experiments than the simulated environment.

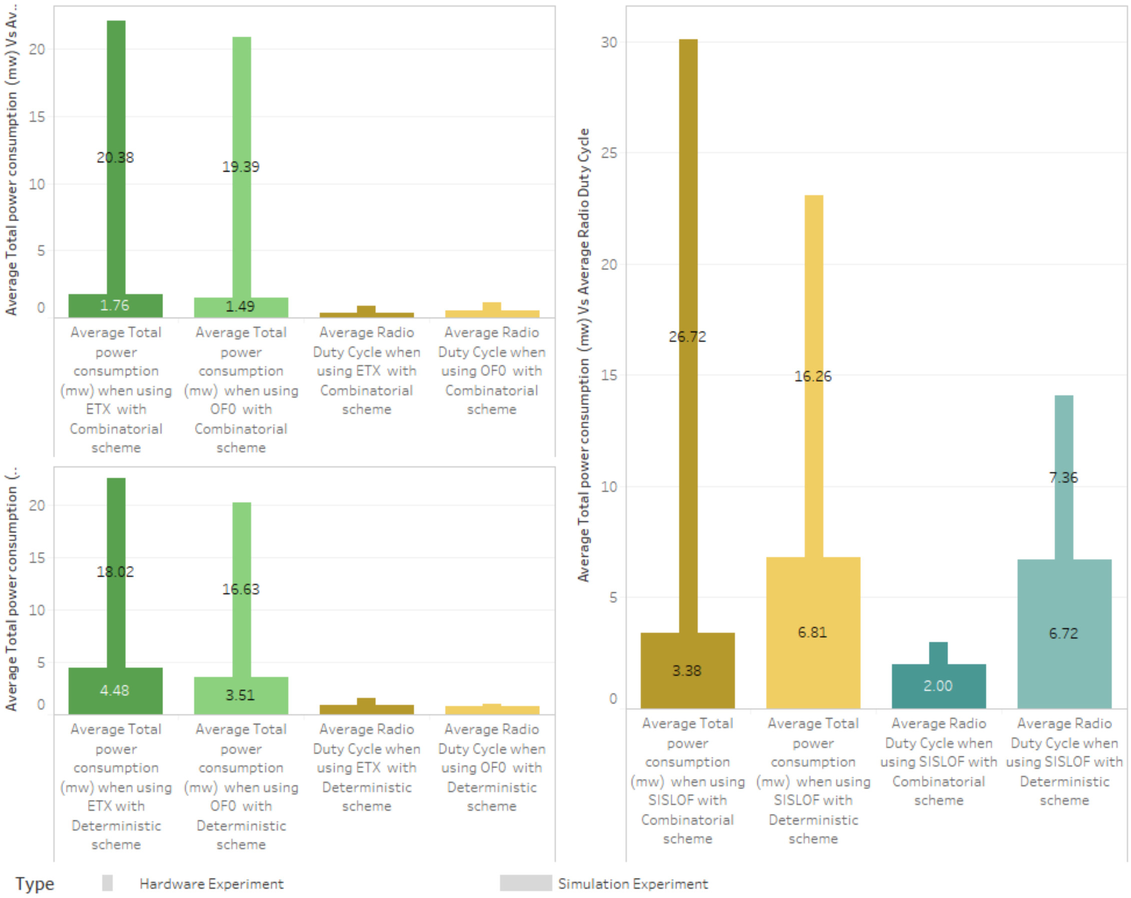


Figure 7.10: Average power consumption and the average radio duty cycle when using the Probabilistic and Deterministic schemes for all objective functions. Power consumption for all power states are higher for the physical environment experiments than the simulated environment.

higher than the simulated environment. We conclude that this is due to the smaller distance between nodes in the physical environment in comparison with the simulated environment which results in an increase in the number of neighbours and number of generated messages. When comparing the CPU usage for all objective functions when using the Probabilistic or Deterministic scheme we note that the results are consistent with the results obtained from the various experiments in previous chapters.

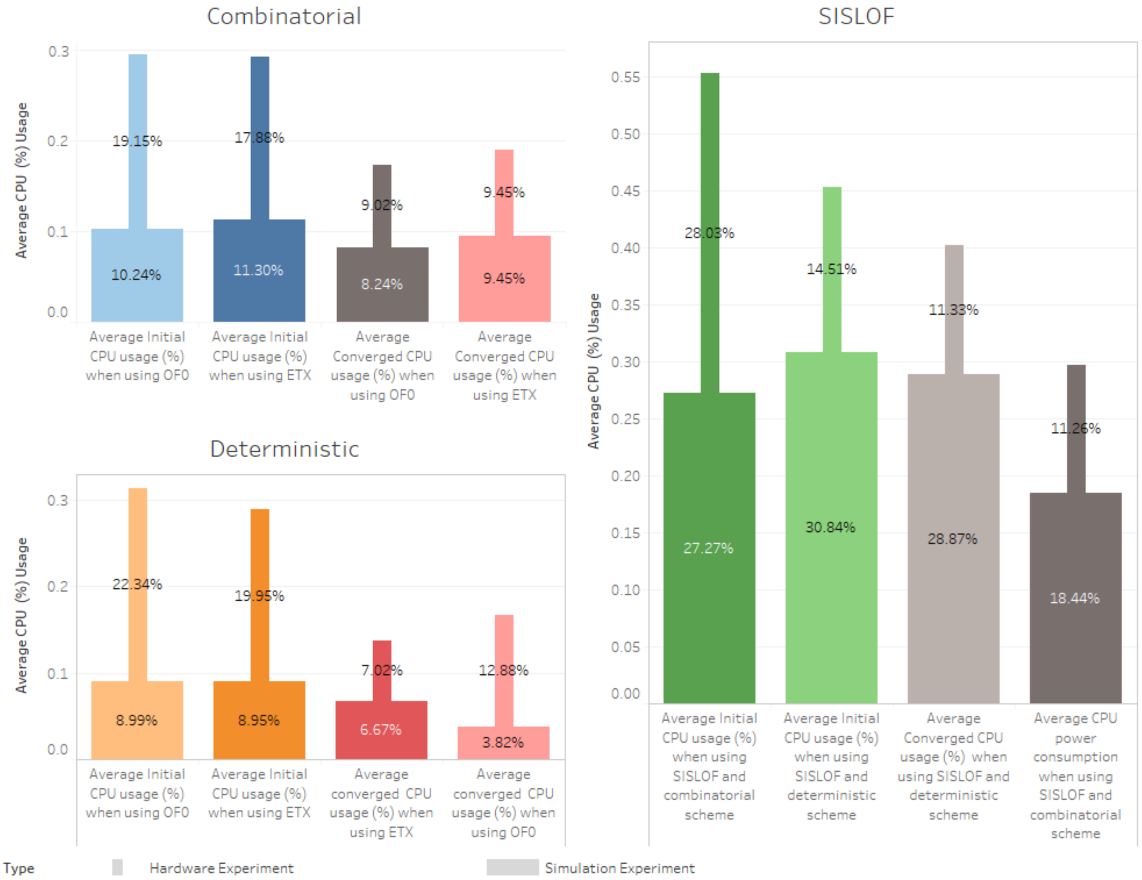


Figure 7.11: Initial and converged CPU usage comparison for both the hardware and the simulation experiments. The CPU usage for is higher for the hardware physical environment than the simulated environment.

7.2.5 Time to Converge

In this section we evaluate the performance of OF0, ETX and SISLOF when using the Probabilistic and Deterministic schemes in the physical environment of 15 nodes. We compare the time it takes for the RPL DODAG to converge in the physical environment with the simulated environment using the same parameters. We observe that the time to converge trends for all experiments return consistent results when comparing networks in a simulated environment and in a physical environment. This is due to the fact that the nodes are spread in a simulated environment more sparsely in comparison with the physical environment where the nodes are spread in a small physical environment such as a room in a building. We also note that for all the objective functions, the Probabilistic scheme outperforms the Deterministic scheme as the time to converge increases when using the Deterministic scheme for both simulated and physical environment experiments. We note that because in smaller networks FMAP authentication has a higher impact on the performance of the network and the computation of the preferred parent does not happen until the authentication occurs. This results in an increase in the TTC for all objective functions.

We also note from the results obtained when comparing the performance of the Objective functions that the results are consistent with the simulated environment. The ETX outperforming OF0 as the preferred parent in OF0 is only based on the rank and the rank of the node changes from when the RPL DODAG formation is initiated until it is formed. This results in several changes for the preferred parent when the rank value changes and it results in an increase in the TTC time for the OF0.

The trends also show that the TTC increased when using the SISLOF objective function in all experiments as the nodes only join the DAG when it identifies that it shares a key with the preferred parent. This puts an overhead on the network and results in a delay to join the network until secured links are identified.

7.2.6 Latency

In this section we compare the latency when using the OF0, ETX and SISLOF objective functions for both the Probabilistic and Deterministic schemes in the physical and simulated environments of 15 nodes and 100 key pools. We note in Figure 7.13 that the latency decreases when running the network in a physical environment. This is due to the nature of the physical environment and the restrictions in space in comparison with the simulated environment.

We also note that the ETX outperforms OF0 for both the Probabilistic and Deterministic schemes when running in the simulated and physical environments. ETX link metrics are an important factor in this case as the link is more stable in comparison with OF0 since in OF0 the preferred parent is chosen based on the rank value of the preferred parent only.

We also observe that the Probabilistic scheme outperforms the Deterministic scheme in all experiments since the FMAP protocol and its voting process results in an increase in the

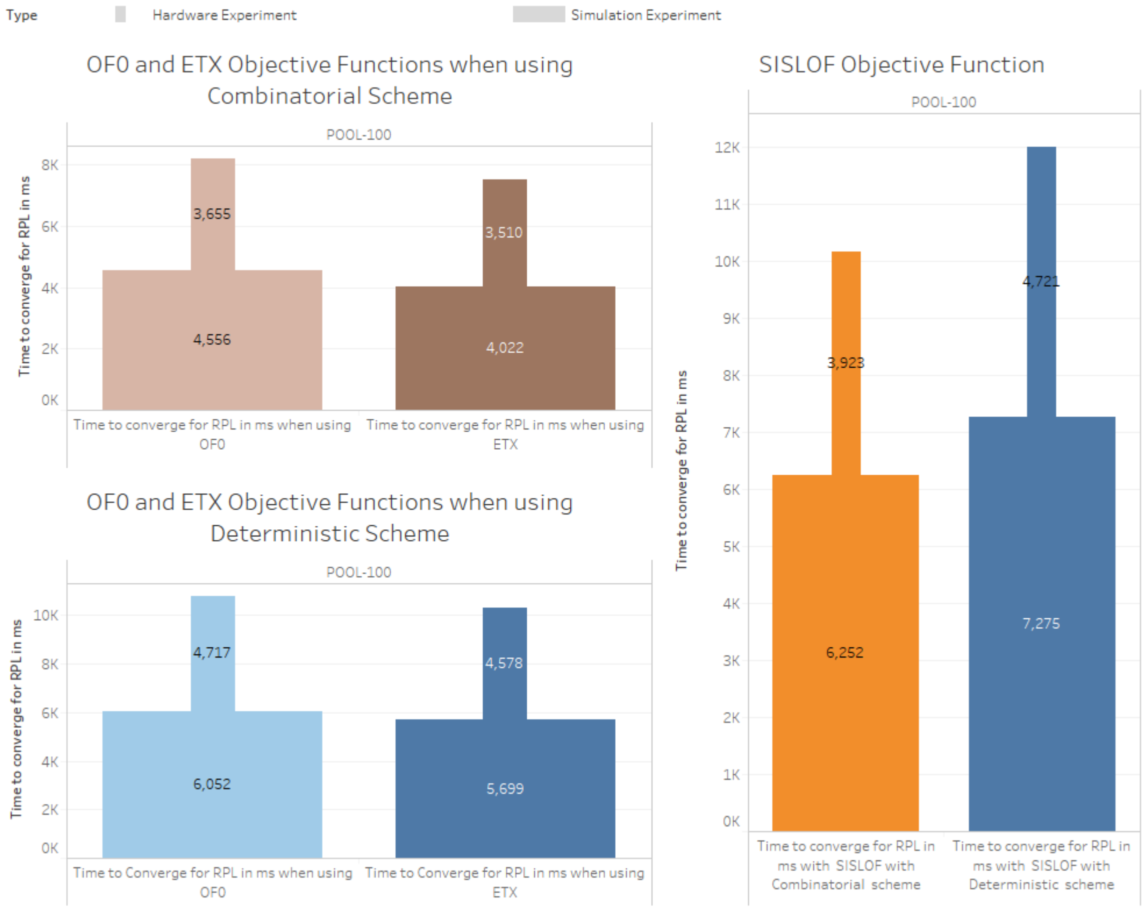


Figure 7.12: RPL DAG average time to converge comparison for both the hardware and the simulation experiments. The time to converge is higher for the simulated environment for all objective functions and when using Deterministic and Probabilistic schemes.

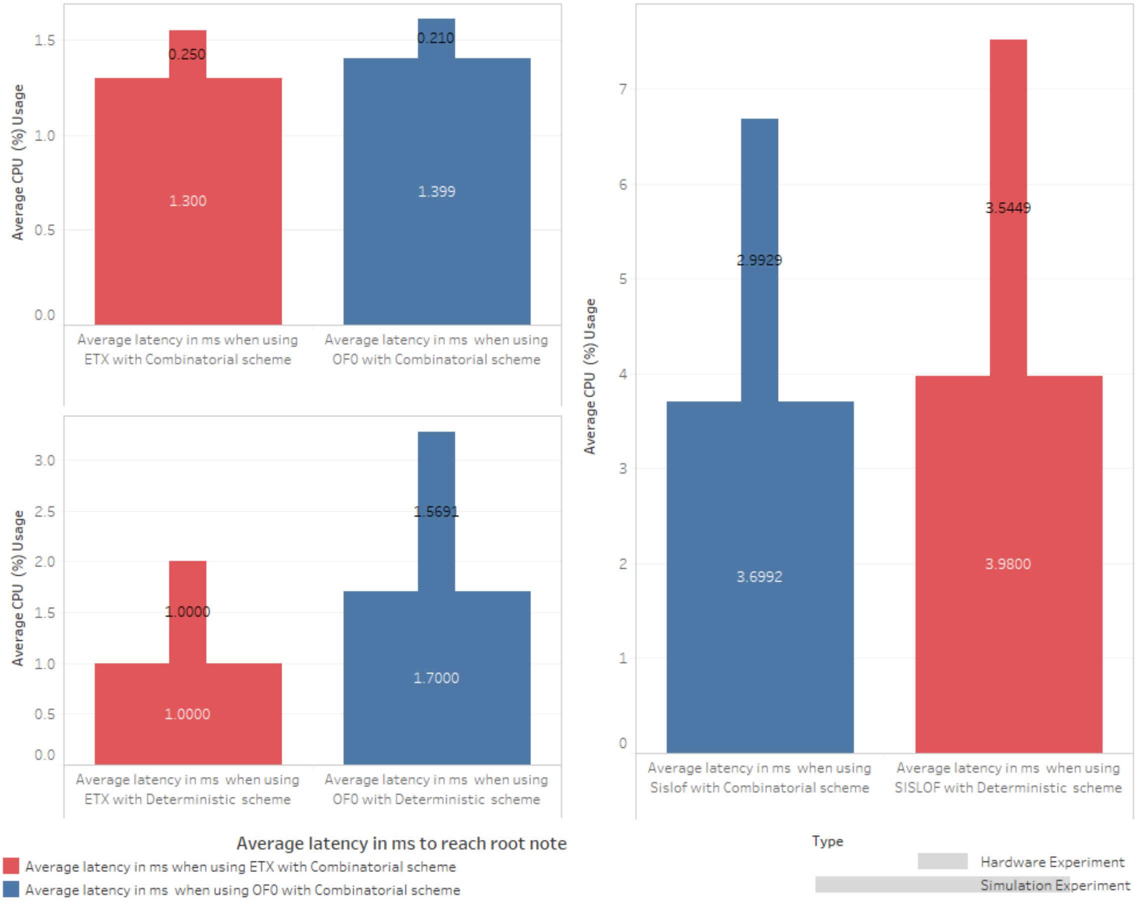


Figure 7.13: Average latency for RPL DODAG comparison for both the hardware and the simulation experiments. Latency is higher for the simulated environment for all objective functions and when using Deterministic and Probabilistic schemes.

number of hop counts as observed in Figure 7.8 which results in an increase in the latency and the time it takes for packets to reach the RPL DODAG root node.

7.2.7 Number of Neighbours

In this section we compare the number of neighbours when using the OF0, ETX and SISLOF objective functions for both the Probabilistic and Deterministic schemes in the physical and simulated environments of 15 nodes and 100 keys pool. We observe in Figure 7.14 that the number of neighbours increases when running the experiments in the physical environment. We note that this is due to the distribution of the nodes in the physical environment and the increase in the number of nodes that are within reach from each other. This results in a physical increase in the number of neighbours.

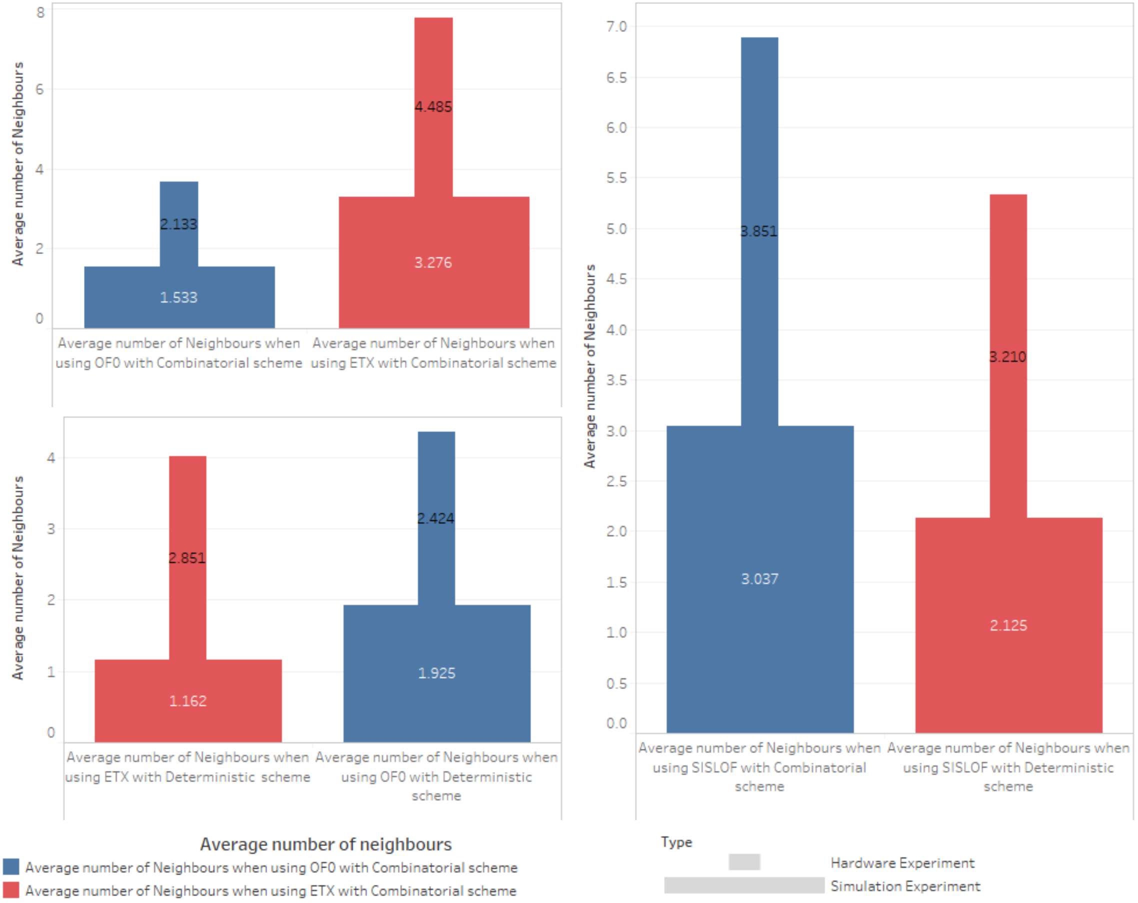


Figure 7.14: Average number of neighbours in a DAG comparison for both the hardware and the simulation experiments. The number of neighbours is higher for the simulated environment for all objective functions and when using Deterministic and Probabilistic schemes.

We also note that the number of neighbours decreases when using the Deterministic scheme in comparison with the Probabilistic scheme and this trend is consistent when using OF0, ETX and SISLOF objective functions. Similarly to previous sections, the FMAP protocol and the mutual trust phase decreases the number of nodes that can be potential preferred parent and are discarded in the first phase.

7.3 Summary

In this chapter we have investigated how the objective functions performs in a real physical environment in comparison with the simulated environment we have experimented with in Chapters 4, 5 and 6. We have also compared the results obtained in those experiments with the results obtained in the simulated environment using the same parameters. We have identified that the results were consistent in term of which objective function outperform the other or which key pre-distribution scheme is more suitable.

The interesting findings however is that the difference between the physical environment and the simulated environment is clearly related to the distance between nodes in the physical environment and the variables of the physical environment that are not present in the simulation environment such as interference, power consumption from battery self discharge decreases, other components power consumption. For this reason, the simulated environment outproduced the hardware environment.

Hardware environment experiments outperformed the simulated one in term of the ring size since the distance between nodes decreased. This naturally resulted in a decrease in storage and processing consumption for the nodes in the physical environment. The physical environment outperformed the simulated environment in both the initial and the converged initial average hop counts because of the distance differences in those environment.

Chapter 8

Analysis

8.1 Introduction

In this chapter we analyse the results obtained in Chapters 4, 5, 6 and 7 and select which key distribution scheme outperforms the other and whether our proposed objective function improves the performance of the nodes and the RPL DODAG formation by generating less overhead in all experiments or whether in some cases the use of other objective functions outperforms SISLOF. We investigate the results obtained in order to justify why our SISLOF objective function outperforms both ETX and OF0 when using the Probabilistic scheme and explain why SISLOF reduces the overhead of the encryption of the traffic between nodes on IoT nodes.

8.2 Ring Size Growth For Full Connectivity

Key findings summary:

- Ring size increase when pool size increase.
- Ring size decrease when network size increase for the same pool size.
- Deterministic scheme outperforms the Probabilistic scheme for ETX and OF0.
- Probabilistic scheme outperforms the Deterministic scheme for SISLOF.
- SISLOF outperforms ETX and OF0 for both Probabilistic and Deterministic schemes.
- OF0 outperforms ETX for both Probabilistic and Deterministic schemes.

In this section, We evaluate the ring size needed to achieve full connectivity when using ETX, OF0 and SISLOF in all the experiments carried out in Chapters 4, 5, 6 and 7. Based on the design of the SISLOF objective function and how the DODAG only contains secured DAGs in comparison with ETX and OF0 where the DODAG is formed before a check is done whether the nodes that form the DAG share a key or not and SISLOF is expected to have the smallest ring size. OF0 is also expected to have smaller ring size than ETX since the link metrics variable is not present and nodes only need to check if a shared key exists.

First, we observe in figure 8.1 that the ring sizes when using SISLOF are relatively smaller than the ring size when using ETX and OF0 in both the Probabilistic and Deterministic schemes.

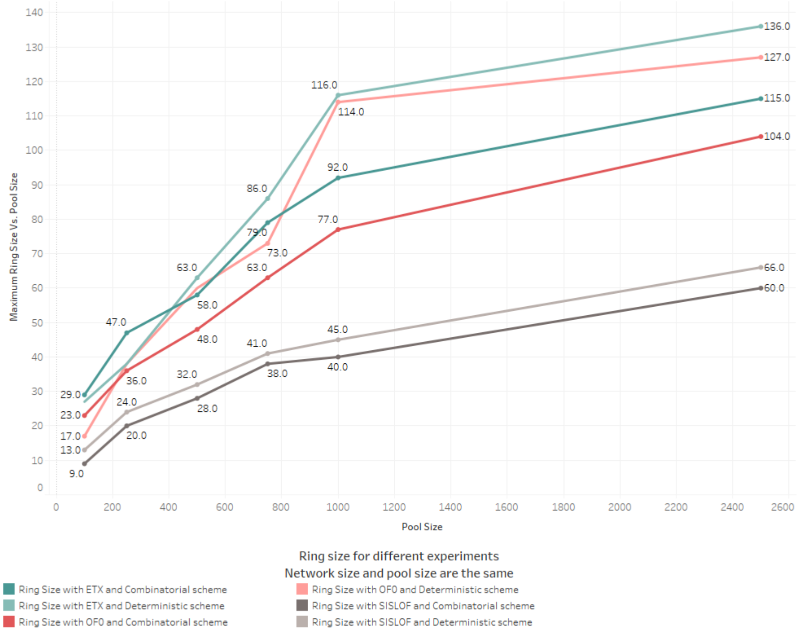


Figure 8.1: Ring size values to achieve full connectivity of network for for all experiments. SISLOF performance in both Probabilistic and Deterministic scheme is clear in comparison with ring sizes achieved when using ETX and OF0 in Chapters 4 and 5.

We also observe in table 8.1 that the ring size increases when the pool size increases. This results applies to all experiments since the probability of two nodes sharing a key decreases when the pool size increases. We also observe that the ring size decreases when the network size increases since the density of the network increases and there are more candidate parents that share a key. The network size factor is clear in the table in the minimum values and maximum values obtained as we note that the minimum values are for larger networks and the maximum values are for smaller networks.

Second, we note that OF0 ring size in all experiments for both the Probabilistic and Deterministic schemes are smaller than the ring size when using ETX. This is due to the fact that OF0 chooses the preferred parent based on the rank only in comparison with the ETX objective function that chooses the preferred parent based on the link metrics and the rank of the node. This added parameter adds a constraint to the selection of a preferred parent and results in less number of nodes that are candidates.

8.3 Ring Size Growth Impact

In this section we analyse the impact of the ring size growth for all experiments in Chapters 4, 5 and 6. For each measurable variable in term of link quality, shared keys between nodes, and the performance of the nodes when the ring size increases or decreases for each objective function.

8.3.1 RPL Control Messages Number

Key findings summary:

- DIO messages increase when pools size increase.
- OF0 and ETX outperforms SISLOF in terms of the number of DIO messages.
- The Probabilistic scheme outperforms the Deterministic scheme for ETX and OF0 objective functions.
- The Deterministic scheme outperforms the Probabilistic scheme for SISLOF objective function.
- DAO messages increase when pools size increase.
- ETX outperforms OF0 in terms of the number of DAO messages.
- OF0 and ETX outperform SISLOF in terms of the number of DAO messages.
- The Probabilistic scheme outperforms the Deterministic scheme for the ETX and the OF0 objective functions.
- The Deterministic scheme outperforms the Probabilistic scheme for SISLOF objective function.
- DIS messages increase when pools size increase.

Ring Size		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	34.8	43.5	61.7	98.0	120.2	143.5
	Min	23.0	36.0	48.0	63.0	77.0	104.0
	Max	47.0	51.0	75.0	135.0	153.0	197.0
ETX [4]	Agv	36.8	50.5	67.3	101.8	129.6	152.5
	Min	29.0	47.0	58.0	79.0	92.0	115.0
	Max	51.0	54.0	81.0	121.0	159.0	212.0
OF0 [5]	Agv	26.8	55.0	69.7	109.8	146.6	166.3
	Min	19.0	49.0	61.0	86.0	116.0	136.0
	Max	31.0	61.0	83.0	141.0	197.0	214.0
ETX [5]	Agv	20.4	49.5	65.7	103.0	134.2	153.8
	Min	17.0	38.0	55.0	73.0	114.0	127.0
	Max	26.0	61.0	78.0	138.0	164.0	198.0
SISLOF [4]	Agv	22.0	22.0	35.0	46.3	50.2	81.8
	Min	12.0	20.0	28.0	38.0	40.0	60.0
	Max	31.0	24.0	41.0	53.0	61.0	96.0
SISLOF [5]	Agv	24.2	25.5	39.3	50.5	54.8	87.7
	Min	17.0	24.0	32.0	41.0	45.0	66.0
	Max	35.0	27.0	46.0	58.0	66.0	104.0

Data labels – From low to high

Average 🟧 Minimum 🟦 Maximum 🟩

Table 8.1: Rings size comparison for all experiments. The ring size for SISLOF for both Probabilistic and Deterministic schemes outperforms OF0 and ETX and have the lowest values in comparison with the other experiments.

- ETX outperforms OF0 in terms of DIS messages.
- OF0 and ETX outperforms SISLOF in terms of the number of DIS messages.
- The Probabilistic scheme outperforms Deterministic scheme for the ETX and the OF0 objective functions.
- The Deterministic scheme outperforms Probabilistic scheme for SISLOF objective function.
- OF0 and ETX outperforms SISLOF in terms of the total number of control messages.
- The Probabilistic scheme outperforms the Deterministic scheme for the ETX and the OF0 objective functions.
- ETX outperforms OF0 in terms of the packets delivery ratio for both the Probabilistic and Deterministic schemes.
- ETX and OF0 outperforms SISLOF in terms of the packets deliver ratio for both Probabilistic and Deterministic schemes.
- The Probabilistic scheme outperforms the Deterministic scheme for all objective functions.

In this section, we evaluate the number of control messages for ETX and OF0 objective functions obtained in both Chapter 4 when using the Probabilistic scheme and in Chapter 5 when using the Deterministic scheme and when using SISLOF for both schemes as shown in Chapter 6. The evaluation of the data presented in Tables 8.3, 8.2 and 8.4

The trends in the data suggests that the number of packets in SISLOF was significantly higher than the number of packets for OF0. In SISLOF, the identifier rings are divided into several parts depending on how big the ring is. This means that for each part of the ring, a DIO message will be sent and a DAO reply will be received. This explains the higher number of messages.

When comparing ETX, OF0 and SISLOF for both schemes we note in Table 8.2 that CMS DIO OF0 < CMS DIO ETX < CMS DIO SISLOF. The nature of the implementation of SISLOF forces the DAG formation to ensure that only leaves that share a key can join the RPL and therefore the number of DIO messages required between nodes is higher as all nodes will be sent a DIS looking for a neighbour that shares a key. OF0 objective function has the smallest number of DIO as OF0 objective function identifies the preferred parent without considering the link metrics that ETX computes. This contributed to the decrease in the number of DIO messages.

First, we observe in Table 8.2 that for ETX and OF0 objective functions, the number of DIO messages when using the Probabilistic scheme are lower than when using the Deterministic scheme. This is because in the first phase of the Deterministic scheme each node attempts to discover which nodes are within its neighbourhood and to verify their identities in order to perform the fingerprinted mutual authentication with each neighbour. However, the FMAP protocol with neighbours in both ETX and OF0 can occur for nodes that do not share a key.

When looking at the number of DIO messages when using SISLOF, we observe that the results of DIO messages for SISLOF are also significantly lower when using the Deterministic scheme rather than when using the Probabilistic scheme. This is due to the fact that in addition of the overhead of the FMAP protocol in SISLOF for the Deterministic scheme, it only contributes to the exchange of DIO messages between nodes that share a key.

When comparing DAO messages in Table 8.3 we notice that the number of DAO control messages is higher for SISLOF in both Probabilistic and Deterministic scheme than when using ETX and OF0. DAO is a response to the DIO message received from the neighbour node and contains the parameters needed to choose the preferred parent.

When comparing ETX, OF0 and SISLOF for both schemes we note in Table 8.3 that CMS DAO ETX < CMS DAO OF0 < CMS DAO SISLOF. The number of DAO control messages for OF0 is larger than ETX since OF0 objective function computes the preferred parent based on the rank of the node which essentially means that more responses will be generated in the response to the DIO received. However, ETX calculates the preferred parent based on the link metric and not only the rank therefore more nodes are discarded from being considered preferred parent and this will result in a lower number of DAO messages. This applies to both experiments when using both the Probabilistic and Deterministic schemes.

When using the SISLOF objective function in both Probabilistic and Deterministic schemes the DAO control messages are significantly larger than the ones for ETX and OF0 similarly to the DIO messages. First due to the design constraints of SISLOF and the limitation due to the size of the packet nodes sent and since SISLOF sends the identifiers ring by the DIO message, the node is forced to respond to each by sending a DAO message with a declaration that a shared key exists or not. This naturally will result in a considerably larger number of DAO messages in comparison with ETX and OF0.

DIO		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	14554.5	78386.4	112895.1	168282.7	217614.0	398216.3
	Min	44.0	43379.2	48065.4	45042.6	52582.0	46317.8
	Max	39125.4	113393.5	193628.7	290537.4	387481.2	967978.5
ETX [4]	Agv	11243.8	58722.9	87138.5	125454.7	168050.6	271658.7
	Min	39.6	33364.4	39680.4	34179.0	42661.2	37546.8
	Max	29858.6	84081.4	147780.1	221118.0	294921.2	736885.4
OF0 [5]	Agv	12408.2	82009.4	130293.4	172364.6	230001.3	421174.8
	Min	42.9	44262.6	61118.3	51197.8	58596.2	64219.5
	Max	54163.0	119756.2	207280.6	294104.1	394679.1	986953.8
ETX [5]	Agv	6769.7	61247.8	97145.6	132724.3	175419.0	294876.8
	Min	38.6	33460.4	40277.8	38761.1	43369.1	39046.9
	Max	19748.4	89035.3	172825.2	234666.9	343824.8	840593.3
SISLOF [4]	Agv	21862.6	117597.7	169381.0	252450.6	326448.7	597345.9
	Min	81.8	65086.9	72146.5	67609.6	78881.3	69486.2
	Max	58703.2	170108.5	290494.8	435824.4	581244.9	1451985.5
SISLOF [5]	Agv	16156.1	106632.6	166736.0	224106.1	299031.0	547561.8
	Min	21.6	57558.8	71462.0	66579.5	76184.7	83507.6
	Max	70461.3	155706.3	269472.2	382366.7	513091.8	1283091.6

Data labels – From low to high

Average 🟧 Minimum 🟦 Maximum 🟩

Table 8.2: Number of DIO messages comparison for all experiments.The highest number of DIO messages are produced when using Probabilistic scheme for ETX and OF0. SISLOF has the lowest number of DIO messages since security is incorporated in the DIO messages and hence many neighbours are discarded before the messages are generated.

We also note in Table 8.3 that for ETX and OF0 objective functions, the number of DAO messages when using the Probabilistic scheme are lower than when using Deterministic scheme. This is for the same reason explained before to justify to the number of DIO messages.

Similarly to the number of DIO control messages, the number of DAO control messages when using the Probabilistic scheme and SISLOF is higher than when using the Deterministic scheme and SISLOF. The FMAP protocol is also contributing to the increase in the number of DAO control messages.

The DIS messages trends in Table 8.4 suggest that that CMS DIS ETX < CMS DIS OF0 < CMS DIS SISLOF. The data reveals that the number of DIS for OF0 is higher since OF0 considers all neighbours potential preferred parents as the rank is the only variable used to compute it in comparison with the ETX objective function that do not consider all neighbours potential preferred parent and some neighbours are discarded for low link metric values. However the number of DIS messages increases when using the Deterministic scheme is higher than when using the Probabilistic scheme since all neighbours that are considered as potential preferred parent participate in the FMAP mutual authentication before checking if nodes share keys or not. This resulted in an increase in the DIS overhead.

The total control messages in Table 8.5 generated over 24 hours for all experiments reflects the number of control messages DIO, DAO and DIS.

It is worth noting that the total of control messages when using SISLOF does not give a clear indication of which key distribution outperforms the other when comparing the Probabilistic and Deterministic schemes.

The packets delivery ratio (PDR) shown in Table 8.6 obtained in the previous results suggests that the packets delivery ratio for all experiments regardless of which objective function and key pre-distribution scheme decreases when the pools size increases. This is directly related to the number of shared keys and to the total control messages generated by all nodes. The PDR results provide a good fit to the expected values as when the total number of control messages increases the number of packets drops increases. The evaluation of the packets delivery ratio obtained from all experiments also suggests that the PDR values for OF0 is lower than the ETX for both the Probabilistic and Deterministic scheme for similar reasons as explained before as OF0 only uses the rank to compute the preferred parent in comparison with ETX that uses link metric to compute the preferred parent and therefore the overhead of the control messages for OF0 is higher which results in an increase in the packets dropped.

DAO		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	4025.9	22462.1	31196.0	44821.5	57803.7	98599.6
	Min	3.5	14317.4	18004.9	15721.4	19375.1	17153.3
	Max	10483.8	30606.8	50204.6	75145.9	100141.2	250495.6
ETX [4]	Agv	3158.2	18925.7	25326.3	36817.8	47429.4	79211.2
	Min	3.2	11351.4	15650.7	12871.5	17106.7	13904.6
	Max	8397.8	26500.0	40221.1	60138.3	80470.2	200165.3
OF0 [5]	Agv	3716.8	25945.0	36681.5	46635.1	61837.2	105071.2
	Min	7.4	15089.0	23318.9	18288.3	21979.2	24254.9
	Max	14956.4	36801.0	54238.3	76317.7	103354.5	255111.5
ETX [5]	Agv	3734.5	21648.9	27641.8	40107.3	51587.7	88909.9
	Min	7.1	12912.2	18788.4	16524.7	19558.6	15619.1
	Max	8476.9	30385.7	41770.9	65662.0	96344.3	230207.4
SISLOF [4]	Agv	5270.5	29229.5	40591.0	58296.9	75185.0	128205.8
	Min	15.9	18635.5	23436.2	20469.4	25238.4	22334.2
	Max	13658.2	39823.4	65302.0	97728.9	130217.5	325664.6
SISLOF [5]	Agv	4854.3	33775.2	47731.0	60657.4	80416.7	136631.7
	Min	13.0	19671.0	30365.8	23829.1	28589.2	31574.7
	Max	19474.0	47879.4	70556.1	99234.1	134413.7	331691.1

Data labels – From low to high

Average Minimum Maximum

Table 8.3: Number of DAO messages comparison for all experiments. The highest number of DAO messages are produced when using Probabilistic scheme for ETX and OF0. SISLOF has the lowest number of DAO messages since security is incorporated in the DAO messages and hence many neighbours are discarded before the messages are generated. SISLOF when using Probabilistic scheme generated less DAO messages than when using Deterministic.

DIS		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	2564.9	14341.2	19725.0	29319.2	37436.9	64267.1
	Min	51.8	8001.9	10871.2	9104.4	12483.7	9886.7
	Max	6587.9	20680.5	32141.7	48336.2	64401.6	160422.1
ETX [4]	Agv	1647.2	9102.7	11728.9	17628.9	22367.6	36848.0
	Min	52.0	5206.8	6468.6	5656.0	6861.6	5854.9
	Max	4121.8	12998.5	19237.3	28331.7	37727.7	93850.9
OF0 [5]	Agv	2601.3	13474.4	23397.4	29081.9	39480.8	66516.3
	Min	45.5	8721.0	14312.9	10805.8	14352.9	14215.9
	Max	9760.3	18227.8	34957.2	46023.9	62716.8	150831.7
ETX [5]	Agv	1216.9	9870.5	13059.8	18685.7	25500.3	40146.2
	Min	44.1	5533.8	6642.9	6510.5	7462.4	5869.2
	Max	2838.2	14207.3	22912.2	30306.8	44459.2	107620.6
SISLOF [4]	Agv	3907.1	21551.4	29620.4	44013.6	56192.7	96430.1
	Min	263.0	12045.2	16347.9	13693.8	18751.0	14845.0
	Max	9890.8	31057.7	48219.9	72556.3	96656.7	240682.9
SISLOF [5]	Agv	3462.1	17538.2	30447.4	37828.9	51356.5	86511.8
	Min	372.1	11355.7	18658.4	14054.6	18695.3	18511.9
	Max	12712.4	23720.8	45469.8	59873.0	81543.3	196107.7

Data labels – From low to high

Average 🟧 Minimum 🟦 Maximum 🟩

Table 8.4: Number of DIS messages comparison for all experiments. ETX objective function outperforms SISLOF and OF0 for both Probabilistic and Deterministic schemes. The probablistic scheme outperforms the Deterministic scheme as the FMAP mutual authentication results in the increase in the DIS.

Experiment		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	21126.6	115189.7	163816.1	242423.4	312854.7	561083.0
	Min	5.4	65698.5	76941.6	69868.4	84440.8	73357.8
	Max	56197.1	164680.8	275975.0	414019.4	552024.0	1378896.3
ETX [4]	Agv	16031.4	86751.3	124193.6	179901.4	237847.6	387717.9
	Min	5.8	49922.6	61799.7	52706.6	66629.5	57306.3
	Max	42378.2	123580.0	207238.5	309588.0	413119.1	1030901.6
OF0 [5]	Agv	18709.9	121428.8	190372.3	248081.6	450384.2	592762.3
	Min	13.8	68072.6	98750.1	80291.8	122870.7	102690.2
	Max	78879.6	174785.1	296476.0	416445.7	808119.1	1392897.0
ETX [5]	Agv	11706.3	92767.3	137847.2	191517.3	408855.1	423932.9
	Min	15.8	51906.4	65709.2	61796.4	123469.2	60535.2
	Max	31063.5	133628.2	237508.3	330635.7	729048.8	1178421.2
SISLOF [4]	Agv	30970.8	168378.6	239592.5	354761.2	525561.6	821981.8
	Min	13.4	95767.5	111930.6	101772.8	94928.2	106665.4
	Max	82252.2	240989.6	404016.6	606109.5	1494750.3	2018333.0
SISLOF [5]	Agv	24395.5	157946.0	247581.0	322592.5	559845.0	770705.3
	Min	21.7	88585.4	128486.2	104463.3	70390.1	133594.1
	Max	102647.8	227306.5	385498.0	541473.8	1079790.4	1810890.5

Data labels – From low to high

Average Minimum Maximum

Table 8.5: Number of total control messages comparison for all experiments. ETX and OF0 outperform SISLOF in term of the number of control messages. Probabilistic scheme outperforms the Deterministic for all experiments.

The PDR for SISLOF when using the Probabilistic scheme is higher than when using the Deterministic scheme. The FMAP authentication when using the Deterministic scheme contributed to the decrease in the PDR since the DAO and DIO messages generated are considerably lower for the Deterministic scheme and this produced a lower PDR values.

8.3.2 Converged And Initial RPL Hop Counts

Key findings summary:

- OF0 outperforms ETX for both Initial and Converged average hop counts for both Probabilistic and Deterministic schemes.
- The Probabilistic scheme outperforms the Deterministic scheme for all objective functions.
- SISLOF outperforms ETX and OF0 when using the Probabilistic scheme.
- OF0 and ETX outperforms SISLOF when using the Deterministic scheme.

In this section we compare the initial and converged average hops count value obtained for all experiments in Chapters 4, 5 and 6 for all objective functions.

The average initial hop count is related directly by the number of nodes in the network and to the ring size. It is expected to have an increase in the hop count when the number of nodes in the network increases. It is also expected for the number of hops to increase if the ring size decreases since the nodes will need to choose a longer path to reach the root node if they share keys with fewer neighbouring nodes.

The average initial average hops count for OF0 is lower than for ETX objective function as shown in Table 8.7 for both the Probabilistic and Deterministic schemes. This is related directly to how OF0 computes preferred parents in comparison with ETX. The trend is sufficient to be considered since the increase in the number of hops is expected both when using ETX over OF0 and when increasing the number of nodes in the network.

The average initial number of hops count for SISLOF when using the Probabilistic scheme is relatively lower than the ones for ETX and OF0. This is due to the fact that OF0 prioritizes a preferred parent node based on the rank of the preferred parent and considers the shared keys the second variable. ETX also prioritize the link metrics over the shared key and this also results in an increase in the average hop counts.

The Deterministic scheme impacts the performance of the average initial hops count for SISLOF in comparison with ETX and OF0 as the FMAP mutual authentication in phase one

PDR		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	89.6	65.0	56.3	51.5	47.2	38.0
	Min	73.0	58.0	46.0	43.0	36.0	18.0
	Max	100.0	72.0	69.0	65.0	62.0	53.0
ETX [4]	Agv	96.6	81.5	78.3	73.3	67.6	58.5
	Min	86.0	79.0	76.0	63.0	56.0	38.0
	Max	100.0	84.0	81.0	78.0	77.0	77.0
OF0 [5]	Agv	91.4	60.0	54.0	49.5	45.6	37.8
	Min	76.0	50.0	42.0	38.0	33.0	16.0
	Max	98.0	70.0	69.0	67.0	67.0	59.0
ETX [5]	Agv	96.2	79.0	73.3	65.5	62.6	54.8
	Min	91.0	75.0	68.0	53.0	49.0	32.0
	Max	100.0	83.0	81.0	79.0	79.0	76.0
SISLOF [4]	Agv	83.6	59.5	51.7	46.8	42.8	33.2
	Min	66.0	53.0	42.0	38.0	30.0	15.0
	Max	93.0	66.0	64.0	59.0	59.0	51.0
SISLOF [5]	Agv	83.8	54.0	47.7	44.3	40.0	31.7
	Min	70.0	42.0	36.0	32.0	27.0	9.0
	Max	90.0	66.0	64.0	64.0	63.0	56.0

Data labels – From low to high

Average 🟧 Minimum 🟦 Maximum 🟩

Table 8.6: Control messages packets delivery ratio comparison for all experiments. The packets delivery ratio when using ETX and OF0 outperforms SISLOF since all packets are discarded before the DODAG formation if nodes do not share a key. Probabilistic scheme outperform Deterministic for all objective functions.

contributes to discarding several nodes and therefore forcing the RPL DODAG formation to choose a preferred parent that has lower rank.

The average converged hop count shown in Table 8.8 using both the Probabilistic and Deterministic schemes when using all objective functions does not reveal a significant difference with the average initial hops count. This result provides a good fit to the expectation in term of the relation with the network size and the number of keys in each ring.

8.3.3 Power Consumption

Key findings summary:

- The Probabilistic scheme outperforms the Deterministic scheme for all objective functions.
- ETX outperforms OF0 and SISLOF when using both Probabilistic and Deterministic schemes.
- SISLOF underperforms ETX and OF0 when using both Probabilistic and Deterministic schemes.
- ETX outperforms OF0 for both Probabilistic and Deterministic schemes.
- The Probabilistic scheme outperforms the Deterministic scheme for all objective functions.

The average total power consumption is related to the number of RPL control messages generated for the RPL DODAG to form. We expect the Probabilistic scheme to outperform the Deterministic scheme as the FMAP protocol will increase the number of RPL control messages. We also expect SISLOF to underperform in comparison with both ETX and OF0 .

The average total power consumption is the sum of all components that consumes power and they are the CPU power consumption to process the control messages, the low power consumption when the radio duty cycle is on, the transmitter power consumption when a node transmits a DAO and the receiver power consumption when a node receives a DIO to form the RPL DODAG. After collecting in Table 8.9 all the experiments results obtained in Chapters 4,5, 6 and 7, we observe that the Probabilistic scheme outperforms the Deterministic scheme for all objective functions as expected. We note that this is due to the increase in the overhead because of the FMAP protocol which increases the number of RPL control messages that are discarded since the nodes failed to agree on a mutual trust. We also note

IAHC		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	6.0	3.8	24.4	24.3	30.2	43.7
	Min	2.1	2.6	15.8	16.2	19.0	28.5
	Max	9.3	4.9	32.4	32.0	42.7	59.8
ETX [4]	Agv	6.9	11.7	27.0	26.8	32.3	47.8
	Min	2.7	9.9	16.8	18.8	21.5	29.7
	Max	9.9	13.5	35.0	35.2	45.3	64.3
OF0 [5]	Agv	4.4	8.1	20.4	23.7	28.3	41.9
	Min	1.9	5.4	11.5	15.5	18.5	24.8
	Max	7.5	10.7	27.1	34.3	43.5	56.9
ETX [5]	Agv	3.6	8.7	28.0	27.8	34.9	48.0
	Min	2.1	6.7	17.4	19.5	23.8	31.7
	Max	4.9	10.6	34.1	38.4	51.5	62.7
SISLOF [4]	Agv	6.3	8.2	18.0	17.3	22.1	36.6
	Min	4.7	7.5	6.8	10.3	11.5	21.7
	Max	7.9	8.9	25.0	27.2	37.3	51.3
SISLOF [5]	Agv	6.8	14.3	32.7	31.5	40.2	52.1
	Min	4.1	13.5	21.7	23.5	28.1	33.7
	Max	8.8	15.0	39.7	42.3	56.9	66.0

Data labels – From low to high

Average Minimum Maximum

Table 8.7: Initial hop counts comparison for all experiments. The Probabilistic scheme outperforms the Deterministic scheme for all objective functions. SISLOF outperforms ETX and OF0 when using the Probabilistic scheme. OF0 and ETX outperforms SISLOF when using the Deterministic scheme.

CAHC		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	6.8	12.7	24.8	29.7	33.8	45.3
	Min	1.9	11.8	17.8	22.1	26.8	36.9
	Max	15.4	13.7	28.2	37.3	44.7	53.3
ETX [4]	Agv	7.9	13.8	25.7	31.8	36.3	51.7
	Min	2.0	12.2	18.3	23.8	26.9	38.4
	Max	17.3	15.3	29.7	41.7	47.3	67.4
OF0 [5]	Agv	3.8	11.9	23.3	27.9	32.6	46.2
	Min	1.3	10.4	15.1	19.9	25.0	33.9
	Max	6.3	13.5	28.4	39.6	47.4	60.6
ETX [5]	Agv	3.3	12.0	30.2	30.5	36.9	51.6
	Min	1.1	10.2	19.0	20.8	25.9	34.6
	Max	8.2	13.9	35.8	42.2	53.9	66.0
SISLOF [4]	Agv	5.9	8.6	19.8	19.1	23.6	38.1
	Min	3.3	7.5	8.0	11.9	12.8	23.3
	Max	8.5	9.7	26.5	29.2	39.0	52.5
SISLOF [5]	Agv	7.9	16.1	35.5	33.9	43.6	55.4
	Min	2.7	14.8	23.4	25.0	31.2	37.2
	Max	10.9	17.4	42.7	43.8	60.9	68.0

Data labels – From low to high

Average Minimum Maximum

Table 8.8: Converged hop counts comparison for all experiments. The Probabilistic scheme outperforms the Deterministic scheme for all objective functions. SISLOF outperforms ETX and OF0 when using the Probabilistic scheme. OF0 and ETX outperforms SISLOF when using the Deterministic scheme.

that the SISLOF objective function underperforms both ETX and OF0 since the secure link parameter is added which adds another overhead to the RPL DODAG formation and causes an increase in the number of RPL control messages that are discarded.

The average radio duty cycle is expected to be low for the Deterministic scheme as there are less trusted nodes in the network hence the number of RPL control messages is less.

The average radio duty cycle obtained for the different scheme experiments and the different objective functions is shown in Table 8.10. We note from those values that the average radio duty cycle (ARDC) that the Probabilistic scheme outperforms the Deterministic scheme. This is due to the fact that when using the Deterministic scheme, less nodes are assumed to be trusted nodes since the FMAP protocol discards nodes that are not trusted. This results in a decrease in the number of times nodes needs to wake up to receive the RPL control messages in comparison with the Probabilistic scheme that does not have this parameter. We also note that SISLOF ARDC is higher for both Probabilistic and Deterministic schemes. The reason the ARDC increases when using SISLOF is because the RPL control messages for both nodes that share a key and nodes that do not share a key are still generated in comparison with the Deterministic scheme FMAP variable that discards the nodes that are not trusted before RPL control messages are generated. It is also noted that ETX outperforms OF0 since the ETX link metric used to form the RPL DODAG results in a more stable network with less RPL control messages.

8.3.4 CPU Usage

Key findings summary:

- OF0 outperforms ETX for Initial CPU usage when using both the Probabilistic and Deterministic schemes.
- OF0 and ETX outperforms SISLOF for Initial CPU usage when using both the Probabilistic and Deterministic schemes.
- The Probabilistic scheme outperforms the Deterministic scheme for Initial CPU usage for ETX and OF0 objective functions.
- It is not sufficient to determine the CPU usage when using SISLOF for both Probabilistic and Deterministic schemes as the hop counts has a high impact on both and results in nearly similar values for the initial CPU usage.
- ETX outperforms OF0 for converged CPU usage when using both the Probabilistic and Deterministic schemes.

Power		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	5.1	14.8	18.7	32.7	36.1	49.0
	Min	2.0	10.5	10.7	15.2	16.0	17.0
	Max	9.0	19.1	26.3	53.6	59.0	75.6
ETX [4]	Agv	4.3	11.7	16.7	24.5	29.0	43.0
	Min	2.1	8.9	7.3	9.7	9.5	14.0
	Max	7.0	14.6	26.9	40.3	53.9	71.0
OF0 [5]	Agv	5.6	26.1	35.2	54.2	64.1	82.6
	Min	3.5	18.8	21.6	29.9	31.6	38.9
	Max	8.0	33.5	45.2	85.4	98.2	108.2
ETX [5]	Agv	5.9	20.8	29.7	41.4	59.5	69.0
	Min	4.5	16.4	14.5	18.2	18.2	24.6
	Max	8.7	25.2	46.6	65.9	95.8	107.2
SISLOF [4]	Agv	7.4	15.1	32.3	26.9	52.6	58.4
	Min	3.0	12.8	22.7	12.5	23.1	16.9
	Max	13.1	17.5	45.1	45.8	70.9	103.8
SISLOF [5]	Agv	6.8	28.3	55.2	53.3	92.9	95.0
	Min	3.3	23.7	38.7	24.0	45.0	32.9
	Max	10.7	32.9	78.4	74.3	116.4	144.2

Data labels – From low to high

Average 🟧 Minimum 🟦 Maximum 🟩

Table 8.9: Average power consumption comparison for all experiments. The Probabilistic scheme outperforms the Deterministic scheme for all objective functions. ETX outperforms OF0 and SISLOF when using both Probabilistic and Deterministic schemes. SISLOF underperforms ETX and OF0 when using both Probabilistic and Deterministic schemes.

ARDC		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 4	Agv	2.28	25.48	44.79	55.96	14.59	29.38
	Min	0.54	11.25	39.70	45.43	10.95	18.64
	Max	6.79	39.70	50.88	69.92	24.60	47.93
ETX 4	Agv	2.58	25.63	34.53	39.87	6.94	26.71
	Min	0.32	7.45	20.45	21.44	4.11	13.71
	Max	9.57	43.82	60.84	62.30	10.76	59.62
OF0 5	Agv	1.35	24.65	48.57	52.42	14.42	33.37
	Min	0.11	17.03	40.27	42.07	7.91	17.16
	Max	2.49	32.27	55.39	63.38	19.82	45.62
ETX 5	Agv	2.05	21.77	36.68	45.12	11.96	32.10
	Min	0.66	16.87	27.65	34.95	9.19	14.71
	Max	4.25	26.68	44.20	52.64	17.10	59.63
SISLOF 4	Agv	5.64	36.49	64.32	57.36	25.66	47.59
	Min	1.38	13.09	49.22	14.42	11.43	18.26
	Max	16.82	59.89	78.37	92.31	43.27	81.89
SISLOF 5	Agv	9.04	42.77	93.43	75.59	44.08	80.28
	Min	4.46	20.90	64.91	25.70	27.49	22.97
	Max	20.75	64.64	118.91	98.87	74.84	146.02

Data labels – From low to high

Average 🟧 Minimum 🟦 Maximum 🟩

Table 8.10: Average Average Radio Duty Cycle comparison for all experiments. OF0 and ETX outperforms SISLOF. Probabilistic scheme outperforms the Deterministic scheme for OF0 and SISLOF.

- OF0 and ETX outperforms SISLOF for converged CPU usage when using both the Probabilistic and Deterministic schemes.
- The Deterministic scheme outperforms the Probabilistic scheme for converged CPU usage for ETX and OF0 objective functions.
- It is not sufficient to determine the CPU usage when using SISLOF for both the Probabilistic and Deterministic schemes as the hop counts has a high impact on both and results in nearly similar values for the converged CPU usage.

We evaluate the percentage of CPU usage both during RPL formation (Initial) and 24 hours after the simulation has run (Converged) for ETX and OF0 objective functions obtained in both Chapter 4 when using the Probabilistic scheme and in Chapter 5 when using the Deterministic scheme and when using SISLOF for both schemes as in Chapter 6.

CPU usage is related directly to the number of RPL control messages generated for each DAG formation. The number of RPL control messages also increases when the ring size decreases and the pool size increases. This is due to the fact that the nodes will need for neighbours they share keys with. CPU usage when using SISLOF will be greater than for ETX and OF0 as a greater number of control messages and hop numbers will naturally result in an increase in the CPU usage. Converged CPU usage will be lower when the DAG is formed.

The CPU usage per node is calculated in terms of percentage of usage. We have compared the average CPU usage in all nodes. We collect the CPU usage in two conditions, first until the RPL routing table converge and second for twenty four hours for experiments simulating DSN networks, IoT using RPL ETx objective and SISLOF. We observe that during the time that the RPL routing table is converging, nodes CPU usage consumption was very high when using the SISLOF objective function. This is due to nodes comparing identifiers in rings. CPU usage in ETX is also quite high. We assume that this is directly related to the high number of packets that each node transmitted or received in order to compare the rings and to select the preferred parent. This observation changed when we left simulations running for twenty four hours. CPU usage decreased gradually after 24 hours. DSN and ETX slightly decreased.

The evaluation of the data presented in the Tables 8.11 and 8.12 suggests that the overhead of SISLOF has the highest impact on the initial and converged CPU usage for SISLOF experiment. It is also sufficient from the results to point that the impact of the overhead of the FMAP mutual authentication in Chapter 5 resulted in the increase in the initial CPU usage in comparison with the values obtained in Chapter 4. However, once the

RPL DODAG is formed, the Deterministic scheme outperforms the Probabilistic scheme as the FMAP mutual authentication overhead is not present since neighbours already determined the mutual authentication before the exchange of the RPL control messages with the nodes that share a key.

8.3.5 Time to Converge

Key findings summary:

- ETX outperforms OF0 for how long it take for the RPL DODAG to converge when using both the Probabilistic and Deterministic schemes.
- OF0 and ETX outperforms SISLOF for the TTC when using both the Probabilistic and Deterministic schemes.
- The Deterministic scheme outperforms the Probabilistic scheme for how long it takes for the RPL DODAG to converge in large networks.
- The Probabilistic scheme outperforms the Deterministic scheme for how long it takes for the RPL DODAG to converge in small networks as seen in Chapter 7.

Time to converge for ETX should outperform OF0 as the preferred parent rank when using OF0 will change several times until the preferred parent is identified and the rank for each node is related directly to nodes discovery in the network. The ETX objective function is associated with the link metric with its neighbours and therefore each node can compute its preferred parent faster. The Probabilistic scheme should outperform the Deterministic scheme as the Deterministic scheme adds an overhead to the calculation of the preferred parent associated with the FMAP authentication that happens before the DAG formation can take place. The SISLOF overhead should have the highest factor on the time to converge between all experiments as the overhead of identifying shared nodes while identifying the preferred parent will have an impact on the time it take for the DAG to converge.

We evaluate the time to converge (TTC) for the RPL DODAG for all experiments in Chapters 4, 5 and 6. We observe in Table 8.13 the average, minimum and maximum TTC values obtained for different pool sizes when using OF0, ETX and SISLOF. We observe that the ETX objective function outperforms OF0 and SISLOF. This is consistent with the expected results since the preferred parent rank changes several times from when the DAG formation is initiated until it converges. SISLOF TTC is higher for all experiments than the ETX and OF0 objective functions and this is consistent with the overhead observed when

ICCPU		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	20.47%	55.31%	63.61%	73.75%	45.27%	57.88%
	Min	10.24%	55.14%	61.35%	60.56%	41.60%	52.82%
	Max	31.34%	55.49%	65.47%	86.98%	48.29%	67.00%
ETX [4]	Agv	20.74%	55.44%	67.51%	81.86%	44.70%	60.93%
	Min	11.30%	51.88%	64.60%	67.53%	42.02%	57.00%
	Max	28.51%	59.00%	70.94%	91.99%	46.20%	71.00%
OF0 [5]	Agv	22.63%	54.49%	66.40%	79.71%	45.35%	64.90%
	Min	8.99%	53.61%	61.79%	70.28%	43.37%	60.86%
	Max	39.38%	55.38%	68.95%	89.66%	48.25%	71.45%
ETX [5]	Agv	22.64%	63.30%	67.59%	84.24%	43.93%	66.24%
	Min	8.95%	59.86%	61.61%	71.59%	40.94%	60.58%
	Max	42.95%	66.74%	71.06%	93.75%	45.96%	70.01%
SISLOF [4]	Agv	34.53%	66.31%	79.55%	91.99%	71.64%	69.19%
	Min	27.27%	62.05%	71.25%	67.94%	56.35%	63.04%
	Max	42.94%	70.58%	91.00%	100.00%	79.13%	77.05%
SISLOF [5]	Agv	37.31%	65.77%	76.61%	88.23%	73.28%	69.42%
	Min	14.51%	58.80%	66.69%	66.80%	64.81%	60.10%
	Max	48.64%	72.74%	83.00%	100.00%	80.81%	75.63%

Data labels – From low to high

Average Minimum Maximum

Table 8.11: Average Initial CPU usage comparison for all experiments. The Probabilistic scheme outperforms the Deterministic scheme for Initial CPU usage for ETX and OF0 objective functions.

CCPU		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	11.39%	32.40%	33.68%	43.67%	49.19%	52.01%
	Min	8.24%	23.39%	23.73%	23.57%	23.33%	23.50%
	Max	14.32%	41.42%	45.96%	58.81%	74.39%	74.00%
ETX [4]	Agv	11.87%	29.51%	30.83%	37.49%	46.01%	50.04%
	Min	9.45%	23.50%	24.58%	23.37%	25.87%	24.06%
	Max	14.80%	35.51%	40.05%	42.84%	72.11%	76.00%
OF0 [5]	Agv	9.97%	23.46%	21.71%	19.91%	18.49%	27.38%
	Min	3.82%	21.10%	19.82%	9.13%	11.80%	20.32%
	Max	12.88%	25.83%	22.81%	32.44%	36.57%	41.26%
ETX [5]	Agv	9.99%	21.32%	17.08%	17.61%	23.73%	29.37%
	Min	6.67%	14.36%	15.88%	6.00%	14.98%	15.05%
	Max	14.48%	28.27%	17.98%	27.63%	31.31%	49.23%
SISLOF [4]	Agv	23.89%	52.51%	59.26%	49.59%	60.68%	60.25%
	Min	11.26%	51.47%	51.78%	43.00%	41.66%	41.93%
	Max	33.72%	53.55%	73.00%	56.46%	72.42%	83.96%
SISLOF [5]	Agv	30.77%	46.44%	56.97%	53.70%	60.78%	63.50%
	Min	11.33%	41.99%	48.46%	48.38%	46.24%	37.05%
	Max	41.52%	50.89%	68.26%	60.17%	78.73%	86.49%

Data labels – From low to high

Average 🟧 Minimum 🟦 Maximum 🟩

Table 8.12: Average Converged CPU usage comparison for all experiments. ETX outperforms OF0 for converged CPU usage when using both the Probabilistic and Deterministic schemes. OF0 and ETX outperforms SISLOF for converged CPU usage when using both the Probabilistic and Deterministic schemes.

comparing the number of RPL control messages exchanged as they are higher than the other objective functions which naturally results in a longer time for the DAG to converge.

8.3.6 Latency

Key findings summary:

- ETX outperforms OF0 for latency when using both the Probabilistic and Deterministic schemes.
- ETX and OF0 outperforms SISLOF for latency when using both the Probabilistic and Deterministic schemes.
- The Probabilistic scheme outperforms the Deterministic scheme for latency for all objective functions.

We expect the latency to increase when using the Deterministic scheme in comparison with the Probabilistic scheme since the FMAP mutual authentication and voting process will add an overhead on the RPL DODAG and will increase the number of hops count in the RPL DODAG and therefore latency will increase. We also expect the latency to be greater when using the SISLOF objective function as the hop counts for the RPL DODAG is significantly larger than the other objective functions.

We compare in this section the latency or propagation delay of time it takes for packets to reach the root node in the RPL DODAG for all experiments and their values obtained in Chapters 4, 5 and 6. The data in Table 8.14 reveals significant differences in latency when using Probabilistic over Deterministic schemes as the mutual authentication FMAP and the voting process contributes to the increase in the latency for all objective functions. We observe that the increase is relatively large between the Probabilistic scheme and the Deterministic schemes for all objective functions.

We also observe that the latency when using OF0 increases in comparison with ETX objective function for both Probabilistic and Deterministic schemes. Although the OF0 computes the preferred parent by identifying the rank value for the neighbouring nodes, the ETX objective function computes the preferred parent by identifying the rank value but also by computing the link metrics to identify the best link available for the neighbouring nodes.

The increase in the average hop counts as observed in Sections 4.14, 5.19 and 6.17 is also a factor that will have an impact on the latency in the network as more traffic in the network

TTC		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	14066.72	37349.77	27484.13	31611.05	44857.42	54308.10
	Min	3655.28	27596.95	25662.36	24122.06	22313.04	23473.41
	Max	28154.14	47102.59	29988.81	39023.82	52269.20	81878.84
ETX[4]	Agv	12587.72	33553.64	26745.33	39291.76	44212.39	53480.11
	Min	3510.46	26721.84	25699.98	25424.92	33626.16	22874.11
	Max	26217.20	40385.43	27959.55	46596.07	47934.19	78466.34
OF0 [5]	Agv	9679.84	31954.10	32369.07	36026.96	39390.84	42147.25
	Min	4716.96	27825.12	30145.60	16330.67	35454.21	21506.71
	Max	25146.78	36083.08	36443.78	45028.43	47397.58	57551.26
ETX[5]	Agv	9998.93	30710.68	31437.27	35004.12	39409.32	41907.76
	Min	4577.88	26316.84	28909.07	15672.29	34462.79	20802.56
	Max	28319.33	35104.51	35921.84	43441.69	45435.39	56797.45
SISLOF [4]	Agv	17237.64	42977.59	49595.24	49901.18	57923.50	58513.52
	Min	3922.50	38213.19	47336.94	40799.75	46415.23	27414.69
	Max	32842.72	47741.98	53158.36	59978.33	68708.11	82902.53
SISLOF [5]	Agv	15626.58	38252.87	38448.26	43079.54	47381.04	47799.36
	Min	4720.75	36819.61	31198.16	37087.33	44368.15	25344.61
	Max	30025.05	39686.12	45842.44	46032.76	51560.28	60545.40

Data labels – From low to high

Average 🟧 Minimum 🟦 Maximum 🟩

Table 8.13: Average time to converge in (s) comparison for all experiments. ETX outperforms OF0 for how long it takes for the RPL DODAG to converge when using both the Probabilistic and Deterministic schemes. OF0 and ETX outperforms SISLOF for the TTC when using both the Probabilistic and Deterministic schemes. The Deterministic scheme outperforms the Probabilistic scheme for how long it takes for the RPL DODAG to converge in large networks only.

and more hops will increase in delay for packets to be transmitted and thus to reach the root node.

We also observe that the latency when using the SISLOF objective function is larger than both ETX and OF0 for both schemes since the number of hops count increases and the time it will take for packets to reach the root node and the number of control messages that will also increase the delay in the network due to an increase in the dropped packets.

8.3.7 Number of Neighbours

Key findings summary:

- OF0 outperforms ETX for the number of secured neighbours when using both the Probabilistic and Deterministic schemes.
- SISLOF outperforms OF0 and ETX for the number of secured neighbours when using both the Probabilistic and Deterministic schemes.
- The Probabilistic scheme outperforms Deterministic scheme for the number of secured neighbours for all objective functions

The number of neighbours is only related to the distribution of the nodes both in the physical and simulated environments. The only difference we can expect is related to the mutual authentication and voting process in the Deterministic scheme that will discard nodes that do not meet its trust conditions and therefore the number of neighbours will naturally decrease.

We observe in Table 8.15 that the number of secured neighbours when using Probabilistic scheme is larger than the number of secured neighbours when using the Deterministic scheme. The mutual authentication phase and the voting process to identify neighbouring nodes that are trusted before checking if the nodes share a key may result in many nodes being discarded from being considered a preferred parent.

We also note that the number of secured neighbours when using SISLOF is considerably larger than when using ETX and OF0. This is because the nodes when using the SISLOF objective function computes the preferred parent by selecting first neighbours with shared keys before using a second metric such as the link metric or the rank value to select the preferred parent and to form the RPL DODAG.

LAT		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	1.21	9.57	12.09	12.95	3.40	10.78
	Min	0.21	9.42	10.65	12.16	3.00	9.86
	Max	3.00	9.72	13.30	13.56	4.00	12.08
ETX[4]	Agv	1.03	7.31	10.10	11.63	2.72	8.44
	Min	0.25	6.40	8.88	9.43	2.30	7.49
	Max	2.10	8.22	10.80	13.11	3.20	9.52
OF0 [5]	Agv	3.70	11.46	17.32	16.58	11.07	13.54
	Min	1.57	8.77	16.74	14.85	8.04	9.87
	Max	10.90	14.14	18.35	17.71	13.35	15.43
ETX[5]	Agv	2.60	11.40	13.35	14.60	10.12	11.83
	Min	1.19	9.93	12.78	12.24	8.91	8.80
	Max	6.46	12.87	14.10	16.12	11.83	13.46
SISLOF [4]	Agv	4.93	13.41	16.18	17.51	8.66	15.02
	Min	2.99	12.88	14.33	16.17	7.09	13.60
	Max	7.57	13.94	17.31	19.64	10.70	16.33
SISLOF [5]	Agv	5.60	16.45	22.61	21.07	15.37	18.72
	Min	3.54	13.43	20.87	19.77	10.80	15.13
	Max	7.84	19.47	23.86	22.14	17.75	21.79

Data labels – From low to high

Average 🟧 Minimum 🟦 Maximum 🟩

Table 8.14: Average latency comparison for all experiments. ETX outperforms OF0 for latency when using both the Probabilistic and Deterministic schemes. ETX and OF0 outperforms SISLOF for latency when using both the Probabilistic and Deterministic schemes. The Probabilistic scheme outperforms the Deterministic scheme for latency for all objective functions.

NNE		Pool Size					
		POOL-100	POOL-250	POOL-500	POOL-750	POOL-1000	POOL-2500
OF0 [4]	Agv	2.96	10.44	30.25	36.75	10.04	18.25
	Min	1.53	5.03	27.73	29.96	4.33	4.94
	Max	5.72	15.84	33.35	49.74	27.10	31.90
ETX[4]	Agv	4.57	30.54	42.59	50.33	19.75	36.50
	Min	3.28	13.09	32.21	38.20	8.41	19.33
	Max	7.05	47.99	48.09	78.49	36.66	62.31
OF0 [5]	Agv	2.81	8.41	24.63	28.80	7.31	15.63
	Min	1.93	2.95	20.72	25.67	3.78	3.24
	Max	4.50	13.86	27.14	34.05	16.33	27.58
ETX[5]	Agv	2.58	12.43	27.88	32.92	6.30	17.24
	Min	1.16	5.08	25.61	29.49	2.13	10.00
	Max	4.27	19.78	29.50	41.77	13.50	29.16
SISLOF [4]	Agv	4.24	22.16	32.51	41.26	15.06	26.54
	Min	3.04	6.83	28.35	36.40	5.95	14.94
	Max	6.64	37.48	35.41	54.64	31.80	43.46
SISLOF [5]	Agv	3.40	19.72	30.58	37.48	12.68	22.75
	Min	2.13	6.78	26.93	33.42	4.38	13.65
	Max	5.76	32.67	32.58	48.05	27.55	35.05

Data labels – From low to high

Average 🟧 Minimum 🟦 Maximum 🟩

Table 8.15: Average number of neighbours comparison for all experiments. OF0 outperforms ETX for the number of secured neighbours when using both the Probabilistic and Deterministic schemes. SISLOF outperforms OF0 and ETX for the number of secured neighbours when using both the Probabilistic and Deterministic schemes. The Probabilistic scheme outperforms Deterministic scheme for the number of secured neighbours for all objective functions

8.4 Summary

In this chapter, we have analysed the experiments conducted in Chapters 4, 5, 6 and 7 in term of the variables assessed. We have identified that the ring size for the SISLOF objective function is smaller than the ones for ETX and OF0 and that the impact of having smaller ring size is a positive impact in term of the nodes performance. We have also identified that the use of the Deterministic scheme increases the overhead of the security on the nodes. This is due to the fact that the mutual authentication phase and the voting process adds the overhead on the routing formation in term of the number of RPL control messages and the number of hops. This also results in larger computation overhead to identify preferred parent.

Chapter 9

Conclusion and Future Work

We first show a brief summary of the research contribution. In the next section we discuss in details how each of the contribution identified here was achieved and what it means for the security of the Internet of Things and where it fits in the bigger picture of encryption algorithms and key pre-distribution schemes for the Internet of Things.

Key findings summary:

- Our SISLOF objective function outperforms the ETX and OF0 objective functions

This thesis presented a study on how two key pre-distribution schemes that are usually used to protect data in transmission in distributed sensor networks perform when used in the context of the Internet of Things using an RPL objective function that was developed to ensure all nodes that share a key can join the DODAG.

A validation study was conducted in Chapter 3 to validate the key ring sizes and identifier ring sizes obtained when using the Probabilistic scheme for key distribution introduced in [46] in the context of the Distributed sensor networks to achieve full connectivity. We have identified in this study that regardless of the size of the pool, using the Equation 3.5.1 will provide a ring size that leads to full connectivity of the network. We then calculated the ring size needed to achieve full connectivity for all the pool sizes simulated in the experiments.

In Chapter 4, we identified that the ring size values that achieve full connectivity when using the ring sizes obtained when the experiments are running in a distributed sensor network environment do not achieve full connectivity since nodes do not have links with many nodes and many nodes are discarded since they do not share a key and hence cannot participate in the RPL DODAG. We kept on increasing the values of the ring sizes for all experiments until a full connectivity of the network is achieved. We then evaluated the impact of increasing the ring sizes on the DODAG formation and on the overhead increasing the ring sizes adds on the IoT devices and the network performance. We compared the ring sizes and the overheads when using the OF0 objective function and when using the ETX objective function.

In Chapter 5, we included the FMAP mutual authentication process introduced in [47] when using the Deterministic scheme for key distribution. We first identified that when using the Deterministic scheme the DODAG does not achieve full connectivity when using the ring sizes obtained in Chapter 3 for both OF0 and ETX. We experimented with the ring sizes for all experiments by increasing their sizes until we achieved full connectivity. We then evaluated the impact those ring sizes have on the DODAG topology and the overhead they add on the IoT nodes and the network performance.

We introduced in Chapter 6 the Shared Identifier Secure Link Objective Function (SISLOF) that forces the nodes to only choose the preferred parent based on whether they share a key with them or not. We then evaluated if the ring size obtained in Chapter 3 can achieve full connectivity when using SISLOF for both the Probabilistic and Deterministic schemes. We evaluated the impact of SISLOF on the IoT nodes and on the DODAG and the overhead it adds on the network performance.

In Chapter 7 we experimented in a small physical environment of 15 nodes and we compared the results obtained with the results of a simulated environment with a similar number of nodes. The physical environment restrictions prevented us from mimicking the same simulated environment as the simulated environment location of the nodes are more random. We observed in these experiments that the ring sizes when running the experiments in the physical environment are smaller than the simulated environment for both the Probabilistic and the Deterministic schemes. We also observed that the physical environment values obtained outperforms the simulated environment in most experiments mainly due to the fact that the distance between nodes in the physical environment is smaller. However, we note that the results differences when using Probabilistic and Deterministic schemes are consistent with what was obtained when running the simulations in the simulated environment. The Probabilistic scheme outperforms the Deterministic scheme in the number of keys in the ring and the number of hops, however, it underperforms in term of the overhead the encryption keys add on the nodes.

Several of the threats identified in Section 2.5.6 are prevented assuming that the keys are not compromised. The confidentiality of the data is protected since eavesdropping is rendered useless as the malicious actor is not able to make sense of the communication between the secured link. Man in the Middle (MiTM) attacks are also prevented and the integrity of the data is protected since the malicious actor is not able to tamper with the transmitted data payload. The malicious actor is still able to gather information related to the packet transmitted since the header is sent in plain text.

Routing formation is also protected against the several threats identified in Section 2.5.6 due to the fact that opposite to the secured modes of RPL introduced in [4] and discussed in Section 2.5.1 which focuses on the encryption of traffic after the DODAG was formed and hence does not prevent a malicious node from joining the network, our research focused on using the encryption keys distributed in the formation of the DODAG and hence a malicious node cannot join the network unless it has compromised a genuine node or has succeeded in conducting a cryptanalytic attack and compromised the keys.

Routing topology attacks such as the blackhole attack and the selective forwarding attack are prevented as well by using secure communication not only to send data but also for the control messages of RPL. This allows genuine nodes that share a key with another node in the DODAG to join the network while preventing malicious nodes from joining the network assuming that they do not have a shared key and no key was compromised.

The proposed solution presented in this thesis to only allow nodes that share a key to join the RPL DODAG has presented with an overview on how encryption overhead can present a challenge in low power devices networks such as the Internet of Things networks. Although the performance of the encryption algorithms was out of context of this research, our focus was on how the keys are distributed and the RPL DODAG formation, some important research problems have been identified. In particular, the overhead that the encryption keys and distribution has added on the performance of the IoT network had a great impact on the performance of the network and its nodes and this is without taking into consideration the overhead of the encryption algorithm and the process of encrypting and decrypting data.

The evaluation of the ring sizes in Chapters 4, 5, 6 and 7 for all objective functions presented us with interesting results and we were able to deduce several outcomes in term of the performance of the objective functions and the key pre-distribution schemes used. An interesting outcome that the experiments resulted in is the size of the ring needed to achieve connectivity. Although we have proved that the Internet of Things network requires significantly larger ring sizes than the distributed sensor networks to achieve full connectivity for both schemes, the SISLOF objective function provided a great improvement in terms of the ring size in comparison with the ETX and OF0.

We have demonstrated that the size of the ring decreases when using the Probabilistic scheme for all objective functions since some of the nodes when using the Deterministic scheme are discarded even if they share a key if they are not considered trusted in the FMAP mutual authentication phase. More importantly we have demonstrated that the ring sizes when using SISLOF are considerably smaller than when using ETX and OF0. This is because the SISLOF objective function is developed in a way that the RPL control messages include the mechanism to compare the identifier rings in order to identify if one ore more identifier is shared and hence a key is shared. On the other hand, with ETX and OF0 the identifiers are checked after a DAG link between those nodes is formed. The decrease in the ring sizes in SISLOF results as well in a decrease in the storage space needed to store the identifier and key rings for each node.

The total number of control messages increases when using SISLOF in both the Probabilistic and Deterministic schemes due to the restrictions on which nodes will share a key and all RPL control messages that do not achieve a secure link will be discarded, however, this process adds an overhead on the process of the routing DODAG formation.

The packet delivery ratio for the RPL control messages for all objective functions is higher when using the Deterministic scheme. This is because for each node when using the Deterministic scheme there are a lesser number of neighbour nodes since some nodes are discarded in the first phase of the identity and trust check using the FMAP mutual authentication protocol. Although the number of RPL control messages when using SISLOF is higher than both ETX and OF0, the packet delivery ratio is higher since there are a lesser number of RPL control messages.

The number of hops in the DODAG RPL network increases when using the Deterministic scheme after the DODAG converges for the ETX objective function and decreases when both SISLOF and OF0 objective functions. When using SISLOF objective function with the Probabilistic scheme the number of hops decreases since as the nodes create DAGs with nodes that they share a link with even if they are not considered preferred parent.

The power consumption for all objective functions when using the Probabilistic scheme outperforms the Deterministic scheme. This is due to the increase in the number of RPL control messages which naturally will increase the total power consumption and the radio duty cycle. We also observe that the SISLOF objective function power consumption is higher than the ETX and OF0 in both schemes as the increase in the overhead to form a secured link will result in an increase in the number of RPL control messages and the increase in the control messages will result in the power consumption.

The CPU usage when generating the RPL DODAG is generally higher as the CPU for each node is generating the RPL control messages and processing the computation to identify a shared key however, when the DODAG is formed the CPU usage decreases. SISLOF CPU usage is higher in both the Probabilistic and Deterministic schemes since the number of RPL control messages is higher and the power consumption is higher.

The Deterministic scheme experiments took less time to converge for all objective functions since there are a lesser number of neighbours for each node that are considered trusted.

The evaluation of the data presented in this research for the overhead on the IoT networks and the devices suggests that the collective benefits of using the Probabilistic scheme outweighs the benefits gained when using the Deterministic scheme in terms of the overhead encrypting the RPL DODAG formation adds on the network and the nodes.

For the current work and the investigation in this research and the development of the secure objective function SISLOF, it is sufficient to point out that different circumstances and scenarios can lead us to suggest the use of a specific key pre-distribution scheme. Still, there is not enough evidence to suggest that the use of one scheme outperforms another in all cases.

However, there is enough evidence in all experiments to suggest that if the concern is the overhead of the key pre-distribution scheme or the objective function on the performance of the nodes then the impact of using the Deterministic scheme is higher on the nodes since there is an increase in the number of RPL control messages and the power consumption this increase leads to. The ring sizes when using Deterministic schemes also lead to an increase in the storage needs. In this case, it is sufficient to suggest that the Probabilistic scheme performs better to secure the RPL DODAG formation. The remit of the experiments also lead us to suggest that the benefits of using SISLOF makes it the most suitable to use.

However if the concern in those experiments is the privacy and trust then this can lead us to suggest the use of the Deterministic scheme as the identity of the nodes will be vetted before allowing them to participate in the DODAG and some nodes that are not trusted can be discarded even if they share one or more keys with their neighbours. The overhead of those nodes establishing a DAG with an alternative node in this case is not a concern.

If the DODAG topology in this network changes many times such as in when the nodes are mobile, then the Probabilistic scheme should be used as there are more nodes that can be candidates for preferred parents and less nodes will be discarded due to the lack of trust.

We have concluded in this research that not all the key pre-distribution schemes can be a viable solution to secure the IoT network specially if the overhead and the limitations of the nodes are a concern. We have also determined that the parameters used to secure the formation of the DSN networks are not suitable for the IoT network since the formation of the links is one to one in the IoT network in comparison with the Distributed Sensors Networks that can form more than one link with different nodes. We were also able in this research to develop an objective function that reduces the overhead of the key pre-distribution schemes on the nodes and on the DODAG formation after the DODAG converges.

9.1 Limitations

9.2 Future work

What are they? Can they motivate?

The work in this research proposed a new objective function for RPL protocol. The new objective function, named Shared Identifier Secure Link Objective Function SISLOF aimed to provide a secure communication between all DAGs in a DODAG. For SISLOF to achieve this, keys are distributed between nodes either using Probabilistic key pre-distribution scheme or the Deterministic key pre-distribution scheme. After the keys are distributed to all nodes, identifiers for keys are shared by the DIO messages in order to identify if the nodes share a key. If a shared key is found between more than one neighbouring node, the link metric with the those neighbours is assessed in order to choose the preferred parent.

Since we were interested in the study of the implementation of secure routing establishments using different objective functions and key pre-distribution schemes, the encryption protocols used were out of context of this research, however, more research is needed to identify how different symmetric encryption algorithms will impact the routing formation and the nodes performance and whether the use of different encryption algorithms will change the results obtained in all experiments significantly. Future studies could investigate the association between the performance of the SISLOF objective function and the encryption protocol used. The use of one encryption protocol in this research was essential to provide uniform results between all objective functions and schemes compared however and similarly to any other change in the network, the encryption protocol has a definite impact on the performance of nodes in the network.

Another point that needs further investigation is the existence of malicious nodes in the network before keys are distributed. This research assumed that if a malicious node exist, it became available after keys were distributed and during the key distribution phase, all nodes were considered genuine. This assumption needs to be addressed in future studies in order to evaluate the impact of key rings revocation.

Future studies should also aim to replicate results in a larger scale physical networks. This will enable us to identify whether the impact on larger scale changes when it is subjected to the external factors that exists in a physical environment.

Similarly to the other objective functions that were proposed for the routing protocol RPL and were submitted as RFCs, the aim for the researcher to submit SISLOF as a standard secured objective function since there is no objective function that produces a secure route during route initialization. To do so, the code for SISLOF needs to be published as an open source, preferably integrated within the Contiki operating system as an objective function to provide security.

List of Acronyms

AES Advanced Encryption Standard

AODV Ad Hoc On Demand Distance Vector

API application programming interface

ARDC Average Radio Duty Cycle

ARP Address Resolution Protocol

CAHC Converged Average Hop Count

CCPU Converged CPU usage

DAO Destination Advertisement Object

DAO-ACK Destination Advertisement Object Acknowledgement

DODAG Destination Oriented Directed Acyclic Graph

DAG Directed Acyclic Graph

DSN Distributed Sensor Networks

DIO DODAG Information Object

DIS DODAG Information Solicitation

FMAP Fingerprinted Mutual,Authentication Protocol

DNS Domain Name System

DoS Denial of Service

DDoS Distributed Denial of Service

DSR Dynamic Source Routing

ETX Estimated Transmission Count

IAHC Initial Average Hop Count

ICPU Initial CPU usage

IANA Internet assigned Numbers Authority

IETF Internet Engineering Task Force

IoT Internet of Things

IP Internet Protocol

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

6LoWPAN IPv6 over Low -Power Wireless Personal Area Networks

KPS Key Pre Distribution

LAT latency

LEAP Localized Encryption and Authentication Protocol

LBOF Load Balancing Objective Function

LLN Low Power and Lossy Network

MiTM Man in the Middle

MAC Message Authentication Code

MRHOF Minimum Rank with Hysteresis Objective Function

NDP Neighbour Discovery Protocol

NNC Number of Node Connected

NNCS Number of Node Connected Securly

NNE Number of Neighbours

OCP Objective Code Point

OF Objective Function

OF0 Objective Function zero

OLSR Optimized Link State Routing

PDR Packets Delivery Ratio

PRR Packet Reception Rate

PKI Public Key Infrastructure

PRNG PseudoRandom Number Generator

RDC Radio Duty Cycle

RFC Request for Comments

RKP Random key pre-distribution

RS Ring Size

ROLL Routing Over Low-Power and Lossy

RPL Routing Protocol for Low-Power and Lossy Networks

SISLOF Shared Identifier Secure Link Objective Function

TC Trasnmission Count

TTC Time To Converge

TCM Total Control Messages

TCMS Total Control Messages Secure

TAOF Traffic Aware Objective Function

TCP/IP Transmision Control Protocol/Internet Protocol

WSN Wireless Sensor Network

References

[1] A. E. Hajjar, G. Roussos, and M. Paterson. On the performance of key pre-distribution for rpl-based iot networks. In *Interoperability, Safety and Security in IoT*, pages 67–78, Cham, 2017. Springer International Publishing.

[2] A. E. Hajjar, G. Roussos, and M. Paterson. Securing the internet of things devices using pre-distributed keys. In *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, pages 198–200, 2016.

[3] A. E. Hajjar, G. Roussos, and M. Paterson. Secure routing in iot networks with sislof. In *2017 Global Internet of Things Summit (GIoTS)*, pages 1–6, 2017.

[4] T Winter, P Thubert, and Et.al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, mar 2012.

[5] Mustafa Kocakulak and Ismail Butun. An overview of wireless sensor networks towards internet of things. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1–6, 2017.

[6] Chee-Yee Chong and S.P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.

[7] Aruna Gupta and T. Sasikala. Secure routing protocols for manet-enabled iot. In *2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, pages 1–4, 2021.

[8] P. Satyanarayana, Jampani Ravi, T. Mahalakshmi, V V Satyanarayana Kona, and V. Gokula Krishnan. Performance analysis of dsr and cache customized dsr steering protocols in wireless mobile adhoc networks. In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 1348–1356, 2021.

[9] Jun Yin, Lei Wang, Chen Han, and Yuwang Yang. Nc-olsr: A network coding based olsr multipath transmission scheme for fanets. In *2017 4th International Conference on Systems and Informatics (ICSAI)*, pages 1007–1012, 2017.

[10] Geoff Mulligan. The 6LoWPAN architecture. page 78, 2010.

[11] IEEE Computer Society. 802.15.4 Low Rate Wireless Personal Area Networks (LR-WPANs), 2011.

[12] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni. Anatomy of threats to the internet of things. *IEEE Communications Surveys Tutorials*, 21(2):1636–1675, 2019.

[13] Panagiotis I. Radoglou Grammatikis, Panagiotis G. Sarigiannidis, and Ioannis D. Moscholios. Securing the internet of things: Challenges, threats and solutions. *Internet of Things*, 5:41–70, 2019.

[14] P. P. Joby and P. Sengottuvelan. A survey on threats and security schemes in wireless sensor networks. 2015.

[15] Swaroop Poudel. Internet of things: Underlying technologies, interoperability, and threats to privacy and security. *Berkeley Technology Law Journal*, 31(2):997–1022, 2016.

[16] Tuhin Borgohain, Uday Kumar, and Sugata Sanyal. Survey of security and privacy issues of internet of things, 2015.

[17] V. Drăgoi, T. Richmond, D. Bucerzan, and A. Legay. Survey on cryptanalysis of code-based cryptography: From theoretical to physical attacks. In *2018 7th International Conference on Computers Communications and Control (ICCCC)*, pages 215–223, 2018.

[18] S. Surendran, A. Nassef, and B. D. Beheshti. A survey of cryptographic algorithms for iot devices. In *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pages 1–8, 2018.

[19] M. Abomhara and G. M. Køien. Security and privacy in the internet of things: Current status and open issues. In *2014 International Conference on Privacy and Security in Mobile Systems (PRISMS)*, pages 1–8, 2014.

[20] X. Chen, K. Makki, K. Yen, and N. Pissinou. Sensor network security: a survey. *IEEE Communications Surveys Tutorials*, 11(2):52–73, 2009.

[21] S. Choudhary and N. Kesswani. Detection and prevention of routing attacks in internet of things. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1537–1540, 2018.

[22] L. K. Bysani and A. K. Turuk. A survey on selective forwarding attack in wireless sensor networks. In *2011 International Conference on Devices and Communications (ICDeCom)*, pages 1–5, 2011.

[23] W. Yang, Y. Wang, Z. Lai, Y. Wan, and Z. Cheng. Security vulnerabilities and countermeasures in the rpl-based internet of things. In *2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 49–495, 2018.

[24] A. Raoof, A. Matrawy, and C. Lung. Secure routing in iot: Evaluation of rpl's secure mode under attacks. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.

[25] Linus Wallgren, Shahid Raza, and Thiemo Voigt. Routing attacks and countermeasures in the rpl-based internet of things. *International Journal of Distributed Sensor Networks*, 9(8):794326, 2013.

[26] Anthéa Mayzaud, Rémi Badonnel, and Isabelle Chrisment. A taxonomy of attacks in rpl-based internet. *International Journal of Network Security*, 18(3):459 – 473, 2016.

[27] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai. The impact of rank attack on network topology of routing protocol for low-power and lossy networks. *IEEE Sensors Journal*, 13(10):3685–3692, 2013.

[28] A. Rehman, M. M. Khan, M. A. Lodhi, and F. B. Hussain. Rank attack using objective function in rpl for low power and lossy networks. In *2016 International Conference on Industrial Informatics and Computer Systems (CIICS)*, pages 1–5, 2016.

[29] A. Mosenia and N. K. Jha. A comprehensive study of security of internet-of-things. *IEEE Transactions on Emerging Topics in Computing*, 5(4):586–602, 2017.

[30] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao. A survey on security and privacy issues in internet-of-things. *IEEE Internet of Things Journal*, 4(5):1250–1258, 2017.

[31] P. Perazzo, C. Vallati, D. Varano, G. Anastasi, and G. Dini. Implementation of a wormhole attack against a rpl network: Challenges and effects. In *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 95–102, 2018.

[32] P. Nagrath and B. Gupta. Wormhole attacks in wireless adhoc networks and their counter measurements: A survey. In *2011 3rd International Conference on Electronics Computer Technology*, volume 6, pages 245–250, 2011.

[33] J. Granjal, E. Monteiro, and J. Sá Silva. Security for the internet of things: A survey of existing protocols and open research issues. *IEEE Communications Surveys Tutorials*, 17(3):1294–1312, 2015.

[34] Kim Thuat Nguyen, Maryline Laurent, and Nouha Oualha. Survey on secure communication protocols for the internet of things. *Ad Hoc Networks*, 32:17–31, 2015. Internet of Things security and privacy: design methods and optimization.

[35] A. S. A. Mohamed Sid Ahmed, R. Hassan, and N. E. Othman. Ipv6 neighbor discovery protocol specifications, threats and countermeasures: A survey. *IEEE Access*, 5:18187–18210, 2017.

[36] N. Ahmed, A. Sadiq, A. Farooq, and R. Akram. Securing the neighbour discovery protocol in ipv6 state-ful address auto-configuration. In *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 96–103, 2017.

[37] Sudhakar and R. K. Aggarwal. A survey on comparative analysis of tools for the detection of arp poisoning. In *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, pages 1–6, 2017.

[38] A. Hoehn and Ping Zhang. Detection of replay attacks in cyber-physical systems. In *2016 American Control Conference (ACC)*, pages 290–295, 2016.

[39] B. Chen, D. W. C. Ho, G. Hu, and L. Yu. Secure fusion estimation for bandwidth constrained cyber-physical systems under replay attacks. *IEEE Transactions on Cybernetics*, 48(6):1862–1876, 2018.

[40] Q. Hu and G. P. Hancke. A session hijacking attack on physical layer key generation agreement. In *2017 IEEE International Conference on Industrial Technology (ICIT)*, pages 1418–1423, 2017.

[41] Z. Lu, F. Chen, G. Cheng, and S. Li. The best defense strategy against session hijacking using security game in sdn. In *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 419–426, 2017.

[42] D. Celebucki, M. A. Lin, and S. Graham. A security evaluation of popular internet of things protocols for manufacturers. In *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6, 2018.

[43] K. Zhang, X. Liang, R. Lu, and X. Shen. Sybil attacks and their defenses in the internet of things. *IEEE Internet of Things Journal*, 1(5):372–383, 2014.

[44] R. John, J. P. Cherian, and J. J. Kizhakkethottam. A survey of techniques to prevent sybil attacks. In *2015 International Conference on Soft-Computing and Networks Security (ICSNS)*, pages 1–6, 2015.

[45] Daniel Genkin, Luke Valenta, and Yuval Yarom. May the fourth be with you: A microarchitectural side channel attack on several real-world applications of curve25519. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 845–858, New York, NY, USA, 2017. Association for Computing Machinery.

[46] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM CCS*, pages 41–47, NY, USA, 2002. ACM.

[47] Kevin J Henry. Secure Protocols for Key Pre-distribution , Network Discovery , and Aggregation in Wireless Sensor Networks. 2015.

[48] Silicon Labs, 2013.

[49] Ian F. Akyildiz and Mehmet Can Vuran. *Wireless Sensor Networks*. Wiley, first edition, 2010.

[50] E Whitman. The “secret weapon” of undersea surveillance. undersea warfare.

[51] M. Siller M. Carlos-Mancilla, E. López-Mellado. Wireless sensor networks formation: Approaches and techniques. *Journal of Sensors*, 2016, 2016.

[52] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini. Convergence of manet and wsn in iot urban scenarios. *IEEE Sensors Journal*, 13(10):3558–3567, 2013.

[53] R. Anggarwal and M. Lal Das. Aggarwal_RFID Security inthe Context of “Internet of Things”_2012.pdf. pages 51–56, 2012.

[54] Special Issue on "security and identity architecture for the future internet" References. *Computer Networks*, 57(10):2215–2217, 2013.

[55] P. Ahmadi, K. Islam, T. Maco, and M. Katam. A survey on internet of things security issues and applications. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 925–934, 2018.

[56] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking. A review on internet of things (iot), internet of everything (ioe) and internet of nano things (iont). In *2015 Internet Technologies and Applications (ITA)*, pages 219–224, 2015.

[57] Zach Shelby and Carsten Bormann. *6LoWPAN The Wireless Embedded Internet*. Wiley, first edition, 2007.

[58] Z. Honggang, S. Chen, and Z. Leyu. Design and implementation of lightweight 6lowpan gateway based on contiki. In *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pages 1–5, 2018.

[59] P. K. Kamma, C. R. Palla, U. R. Nelakuditi, and R. S. Yarrabothu. Design and implementation of 6lowpan border router. In *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)*, pages 1–5, 2016.

[60] N. Janicijević, M. Lukić, and I. Mezei. Routing protocol for low-power and lossy wireless sensor networks. In *2011 19thTelecommunications Forum (TELFOR) Proceedings of Papers*, pages 234–237, 2011.

[61] G. Montenegro, N. Kushalnagar, and et.al. Transmission of ipv6 packets over ieee 802.15.4 networks. RFC 4944, September 2007.

[62] A Conta, S Deering, and M Gupta. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 4443, mar 2006.

[63] Stephen E. Deering and Robert M. Hinden. Internet protocol, version 6 (ipv6) specification. RFC 2460, December 1998.

[64] J Hui and P Thubert. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282, sep 2011.

[65] P. Thubert. Objective function zero for the routing protocol for low-power and lossy networks (rpl). RFC 6552, March 2012.

[66] J P Vasseur, M Kim, and Et.al. Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks. RFC 6551, mar 2012.

[67] N Kushalnagar, G Montenegro, and C Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. RFC 4919, aug 2007.

[68] O Gnawali and P Levis. The Minimum Rank with Hysteresis Objective Function. RFC 6719, sep 2012.

[69] Hui, Jonathan W., and et.al. Ip is dead, long live ip for wireless sensor networks. In *Proceedings of the 6th ACM Conference SenSys*, pages 15–28, NY, USA, 2008. ACM.

[70] O Gnawali and P Levis. The etx objective function for rpl. RFC 6719, May 2010.

[71] M. Qasem, A. Al-Dubai, I. Romdhani, B. Ghaleb, and W. Gharibi. Load balancing objective function in rpl. Draft ietf, 2017.

[72] M. Qasem, A. Al-Dubai, I. Romdhani, B. Ghaleb, and W. Gharibi. A new efficient objective function for routing in internet of things paradigm. In *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–6, 2016.

[73] G. Papadopoulos IMT Atlantique D. Dujovne Universidad Diego Portales N. Montavont Alexander TEI of Thessaloniki, R. Koutsiamanis and IMT Atlantique. Traffic-aware objective function. Draft ietf, 2018.

[74] C. Ji, R. Koutsiamanis, N. Montavont, P. Chatzimisios, D. Dujovne, and G. Z. Papadopoulos. Taof: Traffic aware objective function for rpl-based networks. In *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–5, 2018.

[75] Javier Lopez Rodrigo Roman. Integrating wireless sensor networks and the internet: a security analysis. *Internet Research*, 19:246–259, 2009.

[76] Isabelle Chrisment Anthéa Mayzaud, Rémi Badonnel. A taxonomy of attacks in rpl-based internet of things. *International Journal of Network Security, ACEEE a Division of Engineers Network*, 18:459 – 473, 2016.

[77] Eaton's Cooper Power Systems Business M. Dohler CTTC V. Daza A. Lozano Universitat Pompeu Fabra M. Richardson Ed. Sandelman Software Works T. Tsao, R. Alexander. A security threat analysis for the routing protocol for low-power and lossy networks (rpls). RFC 7416, Jan 2015.

[78] R. Housley Vigil Security N. Ferguson MacFergus D. Whiting, Hifn. Counter with cbc-mac (ccm). RFC 3610, September 2003.

[79] Haowen Chan, Adrian Perrig, and Dawn Song. *Key Distribution Techniques for Sensor Networks*, pages 277–303. Springer US, Boston, MA, 2004.

[80] Rodrigo Roman, Jianying Zhou, and Javier Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266–2279, 2013.

[81] Shahid Raza, Simon Duquennoy, Joel Höglund, Utz Roedig, and Thiemo Voigt. Secure communication for the internet of things—a comparison of link-layer security and ipsec for 6lowpan. *Security and Communication Networks*, 7(12):2654–2668, 2014.

[82] P. Varadarajan and G. Crosby. Implementing ipsec in wireless sensor networks. In *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, 2014.

[83] Michael Healy, Thomas Newe, and Elfed Lewis. *Analysis of Hardware Encryption Versus Software Encryption on Wireless Sensor Network Motes*, pages 3–14. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[84] Haowen Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *2003 Symposium on Security and Privacy, 2003.*, pages 197–213, 2003.

[85] Donggang Liu, Peng Ning, and Wenliang Du. Group-Based Key Pre-Distribution in Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(2):11–20, 2008.

[86] Otmane EL MOUAATAMID, Mohamed LAHMER, and Mostfa BELKASMI. A review on key pre-distribution schemes based on combinatorial designs for internet of things security. *International Journal of Engineering and Applied Physics*, 1(1):1–8, Jan. 2021.

[87] S. A. Camtepe and B. Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 15(2):346–358, 2007.

[88] Jooyoung Lee and Douglas R. Stinson. Deterministic key predistribution schemes for distributed sensor networks. In Helena Handschuh and M. Anwar Hasan, editors, *Selected Areas in Cryptography*, pages 294–307, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[89] Qiang Huang, Johnas Cukier, Hisashi Kobayashi, Bede Liu, and Jinyun Zhang. Fast authenticated key establishment protocols for self-organizing sensor networks. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, WSNA '03, page 141–150, New York, NY, USA, 2003. Association for Computing Machinery.

[90] Donggang Liu and Peng Ning. Multilevel tesla: Broadcast authentication for distributed sensor networks. *ACM Trans. Embed. Comput. Syst.*, 3(4):800–836, November 2004.

[91] Maura B. Paterson and Douglas R. Stinson. A unified approach to combinatorial key predistribution schemes for sensor networks. Cryptology ePrint Archive, Report 2011/076, 2011. https://eprint.iacr.org/2011/076.

[92] B. Yener S.A. Camtepe. Key distribution mechanisms for wireless sensor networks: a survey. *Technical Report TR-05-07*, 2005.

[93] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia. Leap+: Efficient security mechanisms for large-scale distributed sensor networks. volume 2, page 500–528, New York, NY, USA, November 2006. Association for Computing Machinery.

[94] George Oikonomou. Contiki-ng operating system, 2020.

[95] Benoit Thebaudea. An introduction to cooja, 2014.

[96] Anuj Sehgal. Rpl border router, 2016.

[97] Zolertia. Z1 features: Quick hardware tour.

[98] Colina et.al. Internet of things in 5 days with zolertia.

[99] ContikiOS, 2020.

[100] George Fishman S. Estimating sample size in computing simulation experiments. volume 18, pages 21–38, 1971.

[101] Katarzyna Kalinowska-Górska and Fernando Solano Donado. Constructing fair destination-oriented directed acyclic graphs for multipath routing. *Journal of Applied Mathematics*, 2014(948521), 2014.

[102] J. Hui and JP. Vasseur. The routing protocol for low-power and lossy networks (rpl) option for carrying rpl information in data-plane datagrams. RFC 6553, March 2012.

Appendix A

Algorithms

A.1 Algorithm

In algorithm A.1 we present the key distribution and how it will be used in the context of the Internet of Things.

INPUT

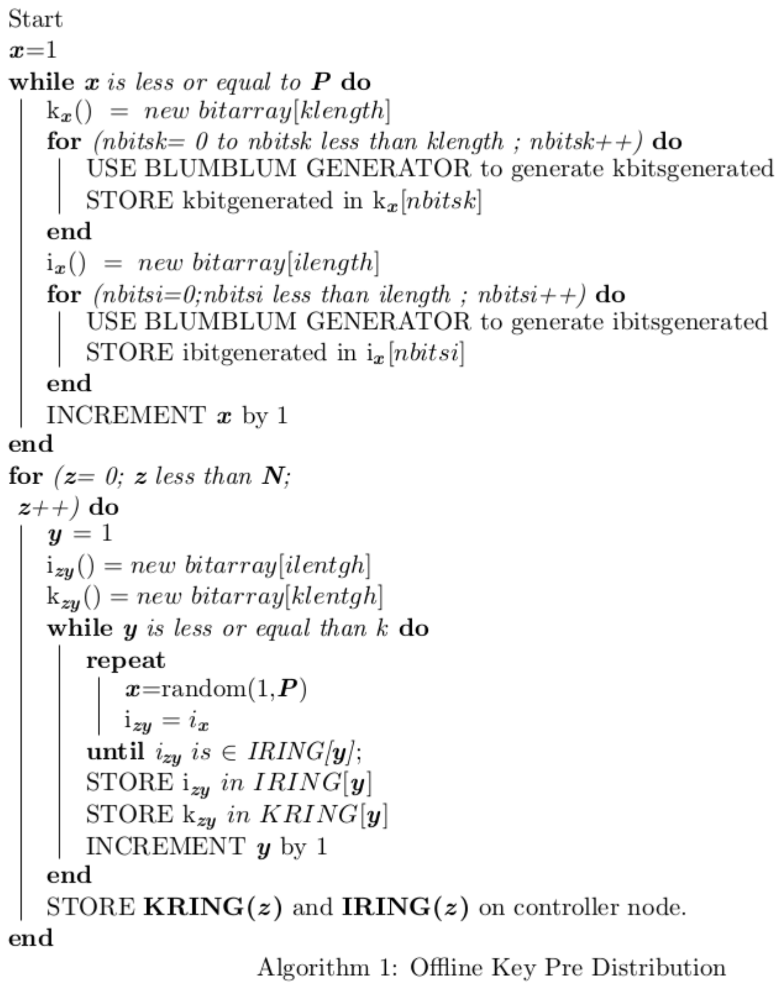
- **P** is the size of the pool, the number of the keys that can be stored in the pool is the size.
- **klength** is a fixed number of bits length for a key.
- **ilength** is a fixed number of bits length for an identifier.
- **N** Number of Nodes in the network.

OUTPUT

- **KRING** is an array of N key rings. Each key ring contains k keys.
- **IRING** is an array of N identifier rings. Each identifier ring contains K identifiers.

$$\mathbf{KRING} = \left[\overbrace{\begin{pmatrix} k_{01} \\ k_{02} \\ k_{03} \\ \vdots \\ k_{0K} \end{pmatrix}}^{0} \overbrace{\begin{pmatrix} k_{11} \\ k_{12} \\ k_{13} \\ \vdots \\ k_{1K} \end{pmatrix}}^{1} \overbrace{\begin{pmatrix} k_{21} \\ k_{22} \\ k_{23} \\ \vdots \\ k_{(2K} \end{pmatrix}}^{2} \dots \overbrace{\begin{pmatrix} k_{(N-1)1} \\ k_{(N-1)2} \\ k_{(N-1)3} \\ \vdots \\ k_{(N-1)K} \end{pmatrix}}^{(N-1)} \right]$$

$$\mathbf{IRING} = \left[\overbrace{\begin{pmatrix} i_{01} \\ i_{02} \\ i_{03} \\ \vdots \\ i_{0K} \end{pmatrix}}^{0} \overbrace{\begin{pmatrix} i_{11} \\ i_{12} \\ i_{13} \\ \vdots \\ i_{1K} \end{pmatrix}}^{1} \overbrace{\begin{pmatrix} i_{21} \\ i_{22} \\ i_{23} \\ \vdots \\ i_{2K} \end{pmatrix}}^{2} \dots \overbrace{\begin{pmatrix} i_{(N-1)1} \\ i_{(N-1)2} \\ i_{(N-1)3} \\ \vdots \\ i_{(N-1)K} \end{pmatrix}}^{(N-1)} \right]$$



```
Start
$x$=1
while $x$ is less or equal to $P$ do
    $k_x()$ = new bitarray[klength]
    for (nbitsk= 0 to nbitsk less than klength ; nbitsk++) do
        USE BLUMBLUM GENERATOR to generate kbitsgenerated
        STORE kbitgenerated in $k_x$[nbitsk]
    end
    $i_x()$ = new bitarray[ilength]
    for (nbitsi=0;nbitsi less than ilength ; nbitsi++) do
        USE BLUMBLUM GENERATOR to generate ibitsgenerated
        STORE ibitgenerated in $i_x$[nbitsi]
    end
    INCREMENT $x$ by 1
end
for ($z$= 0; $z$ less than $N$;
 $z$++) do
    $y$ = 1
    $i_{zy}()$ = new bitarray[ilentgh]
    $k_{zy}()$ = new bitarray[klentgh]
    while $y$ is less or equal than $k$ do
        repeat
            $x$=random(1,$P$)
            $i_{zy} = i_x$
        until $i_{zy}$ is $\in$ IRING[$y$];
        STORE $i_{zy}$ in IRING[$y$]
        STORE $k_{zy}$ in KRING[$y$]
        INCREMENT $y$ by 1
    end
    STORE KRING($z$) and IRING($z$) on controller node.
end
```

Algorithm 1: Offline Key Pre Distribution

Figure A.1: Probabilistic scheme Algorithm.

INPUT

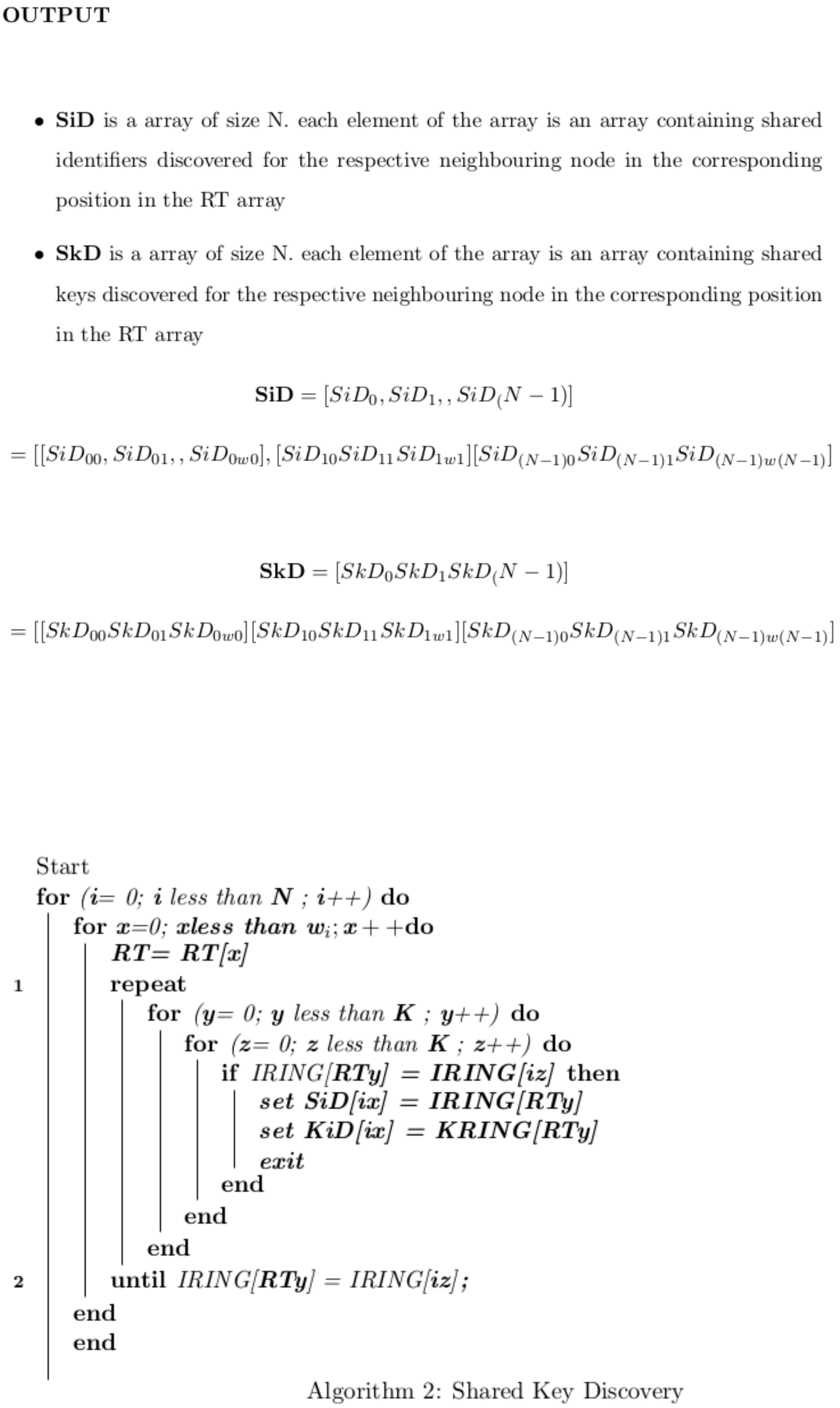
- **KRING** is an array of N key rings. Each key ring contains K keys.
- **IRING** is an array of N identifier rings. Each identifier ring contains K identifiers.
- **RT** is an array that contains list of w_i Neighbouring Nodes RT for each Active node for all N Nodes in the network .

$$\mathbf{KRING} = \left[\overbrace{\begin{pmatrix} k_{01} \\ k_{02} \\ k_{03} \\ \vdots \\ k_{0K} \end{pmatrix}}^{0} \overbrace{\begin{pmatrix} k_{11} \\ k_{12} \\ k_{13} \\ \vdots \\ k_{1K} \end{pmatrix}}^{1} \overbrace{\begin{pmatrix} k_{21} \\ k_{22} \\ k_{23} \\ \vdots \\ k_{(2K} \end{pmatrix}}^{2} \cdots \overbrace{\begin{pmatrix} k_{(N-1)1} \\ k_{(N-1)2} \\ k_{(N-1)3} \\ \vdots \\ k_{(N-1)K} \end{pmatrix}}^{(N-1)} \right]$$

$$\mathbf{IRING} = \left[\overbrace{\begin{pmatrix} i_{01} \\ i_{02} \\ i_{03} \\ \vdots \\ i_{0K} \end{pmatrix}}^{0} \overbrace{\begin{pmatrix} i_{11} \\ i_{12} \\ i_{13} \\ \vdots \\ i_{1K} \end{pmatrix}}^{1} \overbrace{\begin{pmatrix} i_{21} \\ i_{22} \\ i_{23} \\ \vdots \\ i_{2K} \end{pmatrix}}^{2} \cdots \overbrace{\begin{pmatrix} i_{(N-1)1} \\ i_{(N-1)2} \\ i_{(N-1)3} \\ \vdots \\ i_{(N-1)K} \end{pmatrix}}^{(N-1)} \right]$$

$$\mathbf{RT} = [RT_0 RT_1 RT_{(N-1)}]$$

$$= [[RT_{00} RT_{01} RT_{0w0}][RT_{10} RT_{11} RT_{1w1}][RT_{(N-1)0} RT_{(N-1)1} RT_{(N-1)w(N-1)}]$$



OUTPUT

- **SiD** is a array of size N. each element of the array is an array containing shared identifiers discovered for the respective neighbouring node in the corresponding position in the RT array
- **SkD** is a array of size N. each element of the array is an array containing shared keys discovered for the respective neighbouring node in the corresponding position in the RT array

$$\mathbf{SiD} = [SiD_0, SiD_1, , SiD_(N-1)]$$

$$= [[SiD_{00}, SiD_{01}, , SiD_{0w0}], [SiD_{10}SiD_{11}SiD_{1w1}][SiD_{(N-1)0}SiD_{(N-1)1}SiD_{(N-1)w(N-1)}]$$

$$\mathbf{SkD} = [SkD_0SkD_1SkD_(N-1)]$$

$$= [[SkD_{00}SkD_{01}SkD_{0w0}][SkD_{10}SkD_{11}SkD_{1w1}][SkD_{(N-1)0}SkD_{(N-1)1}SkD_{(N-1)w(N-1)}]$$

```
Start
for (i= 0; i less than N ; i++) do
    for x=0; xless than $w_i$; x + +do
        RT= RT[x]
        repeat
            for (y= 0; y less than K ; y++) do
                for (z= 0; z less than K ; z++) do
                    if IRING[RTy] = IRING[iz] then
                        set SiD[ix] = IRING[RTy]
                        set KiD[ix] = KRING[RTy]
                        exit
                    end
                end
            end
        until IRING[RTy] = IRING[iz];
    end
    end
```

Algorithm 2: Shared Key Discovery

Figure A.2: Deterministic scheme Algorithm.

Appendix B

Simulation Data Experiment

Type	Pool	Net	Ch	OF	TCM	PDR	LAT	NNE	TTC	CPUP	CCPU
S	POOL-100	NET-15	4	OF0	6482.307542	100	1.399153294	1.53265	4556.23	0.46671289	0.082353
H	POOL-100	NET-15	4	OF0	5.432659598	98	0.21	2.1325	3655.277534	2.029487574	0.090177276
S	POOL-100	NET-25	4	OF0	14737.40781	96	0.68	2.395557433	12325.32	0.195334969	0.127555411
S	POOL-100	NET-50	4	OF0	28210.55102	81	0.785	3.029837284	21642.65	0.513875917	0.126191601
S	POOL-100	NET-100	4	OF0	56197.13016	73	3	5.719576611	28154.13874	0.460145197	0.143166232
S	POOL-1000	NET-100	4	OF0	150463.1643	62	3.8	6.724536779	22313.04151	0.333840642	0.23327864
S	POOL-250	NET-100	4	OF0	65698.46597	72	3	5.204144684	27596.95162	0.496341175	0.233870556
S	POOL-2500	NET-100	4	OF0	73357.75112	53	4	27.09994068	23473.40682	0.27938063	0.234962019
S	POOL-500	NET-100	4	OF0	138531.8543	69	3	6.847091794	29988.80804	0.123146024	0.237309832
S	POOL-750	NET-100	4	OF0	69868.42991	65	3.2	4.331710423	36768.93904	0.47743518	0.235657487
S	POOL-1000	NET-250	4	OF0	84440.8198	51	9.718046495	15.84066119	52269.20457	0.258652627	0.74391641
S	POOL-250	NET-250	4	OF0	164680.8478	58	9.418604858	5.029876932	47102.58593	0.48	0.414176105
S	POOL-2500	NET-250	4	OF0	201030.2559	50	9.960828654	4.936893388	48099.8551	0.687306751	0.69
S	POOL-500	NET-250	4	OF0	76941.55663	54	9.860809956	12.17957793	25662.35831	0.23399858	0.31358824
S	POOL-750	NET-250	4	OF0	181352.4134	52	10.45440789	16.58762194	26529.37119	0.437420163	0.51689997
S	POOL-1000	NET-500	4	OF0	321120.8121	45	11.37850386	31.89730665	48718.14536	0.56	0.431068488
S	POOL-2500	NET-500	4	OF0	336001.122	44	12.08481352	31.79818765	49370.45644	0.224932406	0.41
S	POOL-500	NET-500	4	OF0	275975.0061	46	10.91874778	12.11303733	26801.23487	0.59	0.459609007
S	POOL-750	NET-500	4	OF0	304453.3383	46	10.65119735	27.72669678	39023.81875	0.521680249	0.406323223
S	POOL-1000	NET-750	4	OF0	456224.4547	42	12.31243059	33.34557697	51456.37106	0.529776956	0.452469609
S	POOL-2500	NET-750	4	OF0	469704.1831	40	13.29529323	29.66422429	56599.47637	0.435117418	0.74
S	POOL-750	NET-750	4	OF0	414019.4344	43	12.15620141	29.96008868	24122.06122	0.717465666	0.588050974
S	POOL-1000	NET-1000	4	OF0	552024.0398	36	12.7087214	33.69102406	49530.32443	0.621615964	0.59883269
S	POOL-2500	NET-1000	4	OF0	907508.1693	23	13.55941014	33.60611085	81878.84343	0.239927855	0.435521711
S	POOL-2500	NET-2500	4	OF0	1378896.26	18	13.37011325	49.73939719	66426.57891	0.342206003	0.61
S	POOL-100	NET-15	4	ETX	5050.379296	100	1.3	3.275699431	4022.32	0.753099579	0.094486725
H	POOL-100	NET-15	4	ETX	5.845233348	100	0.25	4.485083531	3510.46247	2.18	0.094486725
S	POOL-100	NET-25	4	ETX	11025.74561	100	0.7	3.958708489	11323.32	0.21	0.130593999
S	POOL-100	NET-50	4	ETX	21696.95375	97	0.8	4.090189161	17865.31	0.61	0.126191601
S	POOL-100	NET-100	4	ETX	42378.21534	86	2.1	7.054114157	26217.19787	0.63097942	0.147955819
S	POOL-1000	NET-100	4	ETX	118630.6792	71	2.78	17.16636761	33626.16094	0.550823918	0.25873771
S	POOL-250	NET-100	4	ETX	49922.56898	84	2.3	8.410100983	26721.8432	0.66528277	0.234973637
S	POOL-2500	NET-100	4	ETX	57306.2673	62	3.2	36.6643917	22874.10837	0.297642268	0.240613271
S	POOL-500	NET-100	4	ETX	103542.7743	81	2.5	27.17625914	27959.55409	0.19533343	0.245846545
S	POOL-750	NET-100	4	ETX	52706.55677	77	2.8	9.328088175	38782.38297	0.603271905	0.233673648
S	POOL-1000	NET-250	4	ETX	66629.50204	77	8.21745237	47.98534648	45541.48743	0.458375238	0.721085481
S	POOL-250	NET-250	4	ETX	123579.9754	79	6.401457874	13.09177336	40385.42843	0.855595634	0.355126397
S	POOL-2500	NET-250	4	ETX	137159.6094	77	8.229959182	19.33499311	50842.26582	0.755629293	0.72
S	POOL-500	NET-250	4	ETX	61799.70202	78	7.489063882	25.94860441	25699.97802	0.55307458	0.27873748
S	POOL-750	NET-250	4	ETX	129381.9272	78	7.70583206	36.60524227	25424.91719	0.558157717	0.411248519
S	POOL-1000	NET-500	4	ETX	263608.4806	73	9.312312224	42.34596187	46207.87754	0.759294344	0.497637034
S	POOL-2500	NET-500	4	ETX	280210.0926	71	9.517506938	62.30862513	45541.99946	0.230292982	0.46
S	POOL-500	NET-500	4	ETX	207238.4579	76	8.408244614	32.46934648	26576.4436	0.613910398	0.40045331
S	POOL-750	NET-500	4	ETX	227928.9214	75	8.878681554	32.20958602	46596.06929	0.735989004	0.428395266
S	POOL-1000	NET-750	4	ETX	327250.2545	61	10.61909504	48.0948077	47752.21042	0.615288013	0.419902747
S	POOL-2500	NET-750	4	ETX	352942.2326	60	10.79926939	47.47341298	59003.24089	0.555289387	0.76
S	POOL-750	NET-750	4	ETX	309588.0197	63	9.429312673	38.19546902	46363.68417	0.763421498	0.426243502
S	POOL-1000	NET-1000	4	ETX	413119.086	56	11.08235219	41.51983217	47934.19146	0.653186788	0.403100832

S	POOL-2500	NET-1000	4	ETX	467787.7719	43	12.90838073	43.12800144	78466.3359	0.728795551	0.421497121
S	POOL-2500	NET-2500	4	ETX	1030901.594	38	13.10884953	78.48974596	64152.68403	0.389905042	0.400455874
S	POOL-100	NET-15	4	SISLOF	9574.426987	93	3.699153294	3.037250816	6252.000902	0.797534425	0.18441901
H	POOL-100	NET-15	4	SISLOF	13.365544	93	2.992932949	3.850835312	3922.503831	1.97414457	0.11258876
S	POOL-100	NET-25	4	SISLOF	21653.17991	89	6.010064921	4.258014267	13261.21394	0.349557837	0.259421777
S	POOL-100	NET-50	4	SISLOF	41360.68951	77	4.355931863	3.403867406	32842.72288	0.938070383	0.33719027
S	POOL-100	NET-100	4	SISLOF	82252.15551	66	7.570972256	6.639658943	29909.76907	0.72	0.301087797
S	POOL-1000	NET-100	4	SISLOF	220129.5311	59	10.70425089	15.53500442	46415.23365	0.59	0.416607755
S	POOL-250	NET-100	4	SISLOF	95767.51777	66	7.808007991	6.704745307	38213.19419	0.73	0.535519464
S	POOL-2500	NET-100	4	SISLOF	106665.4041	51	9.216410957	31.79608263	27414.69137	0.35	0.59684858
S	POOL-500	NET-100	4	SISLOF	202830.2496	64	8.507049026	15.29202791	48290.41609	0.636353036	0.529993423
S	POOL-750	NET-100	4	SISLOF	101772.7756	59	7.088378942	5.953859745	40799.75221	0.72	0.475463021
S	POOL-1000	NET-250	4	SISLOF	94928.24748	47	13.93601122	37.48019791	64652.61577	0.9	0.695386046
S	POOL-250	NET-250	4	SISLOF	240989.6034	53	12.88210997	6.833042529	47741.97945	0.908693407	0.514727772
S	POOL-2500	NET-250	4	SISLOF	294074.3621	44	13.60277367	15.74242628	53040.46891	0.81	0.41934859
S	POOL-500	NET-250	4	SISLOF	111930.5868	49	15.08237894	16.39083759	47336.9448	0.597174466	0.729974748
S	POOL-750	NET-250	4	SISLOF	265490.7864	49	16.32783295	31.16362926	59978.32994	0.961768335	0.564637243
S	POOL-1000	NET-500	4	SISLOF	337918.5791	43	14.84984478	37.5586561	52479.0833	0.815833318	0.53279213
S	POOL-2500	NET-500	4	SISLOF	491125.5519	38	16.05659369	43.46312978	50501.67259	0.444373534	0.57439904
S	POOL-500	NET-500	4	SISLOF	404016.6227	42	14.21149416	14.94183453	53158.35799	0.71	0.517779908
S	POOL-750	NET-500	4	SISLOF	445671.7371	41	14.33343094	28.35325171	49754.84863	0.89	0.513604032
S	POOL-1000	NET-750	4	SISLOF	480081.2109	35	16.89914303	35.41446164	68708.11421	0.739025529	0.665081129
S	POOL-2500	NET-750	4	SISLOF	687736.6038	34	17.30991361	33.75685877	70259.57102	0.916275382	0.534101172
S	POOL-750	NET-750	4	SISLOF	606109.497	38	16.40840225	36.40085572	49071.80532	0.84	0.429978029
S	POOL-1000	NET-1000	4	SISLOF	1494750.326	30	16.17055614	36.93611205	57362.44857	0.69	0.724192471
S	POOL-2500	NET-1000	4	SISLOF	1333956.149	17	19.64300415	37.04699272	82902.53133	0.803785796	0.650772132
S	POOL-2500	NET-2500	4	SISLOF	2018332.995	15	17.80962625	54.64388559	66962.18744	0.473828258	0.839583444
S	POOL-100	NET-15	5	OF0	3319.166521	98	1.7	1.925492546	6051.689465	1.426647159	0.038182941
H	POOL-100	NET-15	5	OF0	13.80945365	96	1.569098313	2.4236	4716.958788	1.416168326	0.128794
S	POOL-100	NET-25	5	OF0	4205.814933	94	1.918397596	2.631460111	6191.195631	1.456963113	0.10568968
S	POOL-100	NET-50	5	OF0	7131.092396	93	2.421374744	2.556138707	6292.555411	1.634869616	0.120957912
S	POOL-100	NET-100	5	OF0	78879.60333	76	10.90020511	4.498566347	25146.78212	1.401173659	0.104821379
S	POOL-1000	NET-100	5	OF0	182918.3273	67	8.035234315	4.203384853	35631.38968	1.55297406	0.128897777
S	POOL-250	NET-100	5	OF0	68072.57932	70	10.39517371	3.853110241	27825.12406	1.301341378	0.258260993
S	POOL-2500	NET-100	5	OF0	102690.248	59	13.34510923	16.32601813	21506.70706	1.135477671	0.203231415
S	POOL-500	NET-100	5	OF0	175890.6154	69	11.79777506	8.382539624	30145.59874	1.089504228	0.224878632
S	POOL-750	NET-100	5	OF0	80291.84839	67	11.76888459	3.780342846	16330.66762	1.163431538	0.0913
S	POOL-1000	NET-250	5	OF0	122870.6946	53	14.13586629	13.86345561	41016.17312	1.369382931	0.1179702
S	POOL-250	NET-250	5	OF0	174785.1128	50	8.774233612	2.953384291	36083.07759	1.33629087	0.211010198
S	POOL-2500	NET-250	5	OF0	260161.6795	51	9.867943406	3.235324486	31294.97412	1.312571212	0.269879084
S	POOL-500	NET-250	5	OF0	98750.13907	51	15.42931688	10.85144726	36443.7763	1.417611507	0.198225267
S	POOL-750	NET-250	5	OF0	186911.5513	54	15.06900398	13.64625958	38974.22395	1.26826024	0.220745764
S	POOL-1000	NET-500	5	OF0	469846.042	41	14.81096742	26.92598248	37454.86475	1.362168837	0.123092383
S	POOL-2500	NET-500	5	OF0	357515.2224	36	14.90189884	27.58228274	38507.63701	1.45176147	0.23632595
S	POOL-500	NET-500	5	OF0	296476.0156	42	11.1366255	11.54583714	30517.821	1.274011458	0.228060757
S	POOL-750	NET-500	5	OF0	308677.1743	39	18.35308409	20.72016791	45028.43499	1.362857139	0.159885563
S	POOL-1000	NET-750	5	OF0	668166.8693	34	16.86597174	26.04010516	35454.20528	1.211530736	0.18866988
S	POOL-2500	NET-750	5	OF0	498428.3065	36	16.7448644	27.14216287	57551.2646	1.418425478	0.267141124
S	POOL-750	NET-750	5	OF0	416445.7376	38	14.84684495	25.66879036	43774.49975	1.252694012	0.324410153
S	POOL-1000	NET-1000	5	OF0	808119.0944	33	17.03309672	28.04812095	47397.57688	1.409004159	0.365737374
S	POOL-2500	NET-1000	5	OF0	944881.062	29	16.73086659	27.42321369	48157.21852	1.232752499	0.253362826
S	POOL-2500	NET-2500	5	OF0	1392897.044	16	17.71446902	34.04634259	55865.69271	1.195619215	0.41258
S	POOL-100	NET-15	5	ETX	4742.648722	100	1.299153294	1.161813925	5699.153294	1.69974227	0.066687848
H	POOL-100	NET-15	5	ETX	15.81012403	98	1.189270761	2.850835312	4577.884646	1.915384798	0.070248273
S	POOL-100	NET-25	5	ETX	7729.498383	97	1.866336058	2.238645597	5699.153294	1.599686922	0.098469475
S	POOL-100	NET-50	5	ETX	14980.0519	95	2.164895227	2.387368803	5699.153294	1.7342743	0.119263115
S	POOL-100	NET-100	5	ETX	31063.53875	91	6.457623761	4.268467461	28319.32585	1.538967543	0.144782737
S	POOL-1000	NET-100	5	ETX	128159.5828	79	9.779851588	3.960939491	41708.01896	1.793747571	0.149844714
S	POOL-250	NET-100	5	ETX	51906.44619	83	9.47923586	2.12813756	26316.84267	1.363631328	0.143590744
S	POOL-2500	NET-100	5	ETX	60535.20577	65	10.57998418	13.50445811	20802.56028	1.295567211	0.231452638
S	POOL-500	NET-100	5	ETX	110324.1946	81	8.910587142	8.272733487	28909.06534	1.138210917	0.158828151
S	POOL-750	NET-100	5	ETX	61796.38219	79	11.83380789	3.656064543	15672.29093	1.52979094	0.060041733
S	POOL-1000	NET-250	5	ETX	123469.2196	74	12.8659714	19.78308662	39535.95779	1.4377151	0.280219888
S	POOL-250	NET-250	5	ETX	133628.1901	75	9.926607481	5.083085857	35104.50767	1.435132153	0.282736083
S	POOL-2500	NET-250	5	ETX	145462.5616	76	8.795123228	12.92242605	29839.24051	1.449447417	0.150458389
S	POOL-500	NET-250	5	ETX	65709.20703	71	11.2617264	12.52126429	35921.84368	1.579963648	0.173643714
S	POOL-750	NET-250	5	ETX	135955.6541	70	12.07450401	11.12891172	37891.9297	1.480203014	0.20570603
S	POOL-1000	NET-500	5	ETX	439388.2606	62	13.1883431	27.69788404	35904.43498	1.450557722	0.208815951
S	POOL-2500	NET-500	5	ETX	298019.2825	59	13.46244138	29.16035908	41229.01737	1.741467875	0.260176832
S	POOL-500	NET-500	5	ETX	237508.2964	68	12.21498484	10.00378523	29480.88954	1.362195046	0.179847724
S	POOL-750	NET-500	5	ETX	237681.3418	60	13.16638679	25.60641962	43441.69376	1.54655223	0.162347954
S	POOL-1000	NET-750	5	ETX	624209.5126	49	12.78256255	28.53330269	34462.78935	1.532431842	0.234497076
S	POOL-2500	NET-750	5	ETX	358692.3334	51	14.09839109	29.49870763	56797.44985	1.639716073	0.290343088
S	POOL-750	NET-750	5	ETX	330635.6648	53	14.31360173	29.66399729	43010.58406	1.452403232	0.276281076

S	POOL-1000	NET-1000	5	ETX	729048.7598	49	12.24196445	30.76836615	45435.38861	1.580339587	0.313064941
S	POOL-2500	NET-1000	5	ETX	502467.0418	46	15.70770792	29.49289139	46556.74779	1.580905308	0.337374412
S	POOL-2500	NET-2500	5	ETX	1178421.246	32	16.11683906	41.76793286	56221.54857	1.34494478	0.4922645
S	POOL-100	NET-15	5	SISLOF	4411.990266	90	3.98	2.125492546	7274.696438	1.85455641	0.288704325
H	POOL-100	NET-15	5	SISLOF	21.68747814	87	3.5449169	3.209850835	4720.749216	2.014505191	0.113265
S	POOL-100	NET-25	5	SISLOF	5509.459135	87	7.148318397	2.931460111	7213.486629	1.8779877	0.406807646
S	POOL-100	NET-50	5	SISLOF	9386.701491	85	5.476771199	2.956138707	30025.05277	2.011385526	0.314336013
S	POOL-100	NET-100	5	SISLOF	102647.7822	70	7.839186157	5.755663474	28898.92645	1.7	0.415151797
S	POOL-1000	NET-100	5	SISLOF	237904.993	63	14.3394996	13.85144726	44368.15046	1.920443975	0.462370472
S	POOL-250	NET-100	5	SISLOF	88585.43301	66	10.79659618	4.653110241	36819.60865	1.452763496	0.419911186
S	POOL-2500	NET-100	5	SISLOF	133594.1251	56	17.75051462	27.54583714	25344.60922	1.440237598	0.705008874
S	POOL-500	NET-100	5	SISLOF	228758.8778	64	17.32900713	12.95338429	31198.16357	1.927995119	0.484624391
S	POOL-750	NET-100	5	SISLOF	104463.2936	64	16.61571981	4.382539624	37087.3332	2.154786694	0.52558932
S	POOL-1000	NET-250	5	SISLOF	70390.13634	46	19.47029035	32.66879036	46797.8007	1.572780258	0.681984187
S	POOL-250	NET-250	5	SISLOF	227306.4861	42	13.4260788	6.780342846	39686.12344	1.849225086	0.508916202
S	POOL-2500	NET-250	5	SISLOF	338339.3224	43	15.13306837	14.20338485	42085.23601	1.848234638	0.370547761
S	POOL-500	NET-250	5	SISLOF	128486.183	43	20.95142704	13.64625958	45842.43852	1.764745042	0.682600484
S	POOL-750	NET-250	5	SISLOF	243060.3381	47	21.02552213	26.72016791	44930.93104	1.655958197	0.601709189
S	POOL-1000	NET-500	5	SISLOF	1079790.371	34	21.78928026	33.04010516	51560.28085	1.659535243	0.547002534
S	POOL-2500	NET-500	5	SISLOF	464877.0986	29	18.15730743	35.04812095	42281.16711	1.854788564	0.59122062
S	POOL-500	NET-500	5	SISLOF	385497.9691	36	15.23733745	13.86345561	38304.17627	1.575240174	0.541925494
S	POOL-750	NET-500	5	SISLOF	401372.4112	32	23.86092342	26.92598248	46032.75883	1.732881091	0.53684359
S	POOL-1000	NET-750	5	SISLOF	926511.3444	27	20.86727607	32.23532449	45307.62566	1.621970095	0.560551658
S	POOL-2500	NET-750	5	SISLOF	648090.1491	29	23.10100807	32.58228274	58123.55973	1.850194363	0.608432732
S	POOL-750	NET-750	5	SISLOF	541473.782	34	19.77072392	34.32601813	44267.13469	1.749913089	0.483799163
S	POOL-1000	NET-1000	5	SISLOF	484628.3422	30	22.14151164	34.14216287	48871.34409	1.765106173	0.787275222
S	POOL-2500	NET-1000	5	SISLOF	1228440.392	24	21.25690652	33.42321369	60545.4033	1.655955218	0.670130386
S	POOL-2500	NET-2500	5	SISLOF	1810890.505	9	21.11386778	48.04634259	58416.18744	1.782160346	0.864945261

Type	Pool	Net	Ch	OF	LPMP	TxP	RxP	APC	ARDC	ICPU	CCPU
S	POOL-100	NET-15	4	OF0	0.321269376	0.321428571	0.378571429	1.487982266	0.538942513	0.10236	0.082353
H	POOL-100	NET-15	4	OF0	0.581408348	4.408688722	12.36911046	19.38869511	0.577884646	0.19153848	0.090177276
S	POOL-100	NET-25	4	OF0	0.136776163	3.214285714	3.785714286	7.332111132	8.2	0.313374568	0.127555411
S	POOL-100	NET-50	4	OF0	0.08	0.803571429	0.946428571	2.343875917	1.214789077	0.187985659	0.126191601
S	POOL-100	NET-100	4	OF0	0.055492153	1.607142857	1.892857143	4.015637349	6.78771622	0.228021521	0.143166232
S	POOL-1000	NET-100	4	OF0	0.4	6.898	7.9898	15.62164064	13.27182947	0.482949923	0.23327864
S	POOL-250	NET-100	4	OF0	0.384570659	4.656	4.9865	10.52341183	10.94604434	0.436451406	0.233870556
S	POOL-2500	NET-100	4	OF0	0.38504928	7.6568	8.6898	17.01102991	24.6024445	0.416030904	0.234962019
S	POOL-500	NET-100	4	OF0	0.39	5.656	4.23232	10.40146602	12.26057166	0.474219579	0.237309832
S	POOL-750	NET-100	4	OF0	0.29	6.3265	7.6568	14.75073518	11.87140477	0.45372279	0.235657487
S	POOL-1000	NET-250	4	OF0	0.28133119	16.14285714	18.85714286	35.53998382	39.70219021	0.551404546	0.74391641
S	POOL-250	NET-250	4	OF0	0.22	8.035714286	9.464285714	18.2	11.2486391	0.554860928	0.414176105
S	POOL-2500	NET-250	4	OF0	0.22	21.3265	21.3268	43.56060675	33.52480309	0.56	0.69
S	POOL-500	NET-250	4	OF0	0.29	9.6565	8.6568	18.83729858	23.69284196	0.545818627	0.31358824
S	POOL-750	NET-250	4	OF0	0.22	10.26565	12.365	23.28807016	20.91718902	0.528155603	0.51689997
S	POOL-1000	NET-500	4	OF0	0.09	10.6568	11.355	22.6618	47.9295703	0.612503744	0.431068488
S	POOL-2500	NET-500	4	OF0	0.283571525	24.6568	26.989	52.15430393	31.56420982	0.67	0.41
S	POOL-500	NET-500	4	OF0	0.25	11.07142857	13.92857143	25.84	18.63622266	0.556308207	0.459609007
S	POOL-750	NET-500	4	OF0	0.45	18.6565	19.3268	38.95498025	39.70347031	0.613527887	0.406323223
S	POOL-1000	NET-750	4	OF0	0.21	21.3565	22.9896	45.08587696	43.77334586	0.654713981	0.452469609
S	POOL-2500	NET-750	4	OF0	0.345842059	26.35714286	28.64285714	55.78095948	50.88072328	0.64	0.74
S	POOL-750	NET-750	4	OF0	0.324578436	24.10714286	28.39285714	53.5420441	48.07498149	0.6055979	0.588050974
S	POOL-1000	NET-1000	4	OF0	0.21	29.3265	28.35487	58.51298596	45.4270089	0.754713981	0.59883269
S	POOL-2500	NET-1000	4	OF0	0.446220152	23.3265	24.9885	49.00114801	60.41483699	0.72	0.435521711
S	POOL-2500	NET-2500	4	OF0	0.37	37.6598	36.68	75.052006	69.91939068	0.86975	0.61
S	POOL-100	NET-15	4	ETX	0.49	0.25	0.264285714	1.757385293	0.324637954	0.113011761	0.094486725
H	POOL-100	NET-15	4	ETX	0.63	3.84	13.73296267	20.38296267	0.51046247	0.178771265	0.094486725
S	POOL-100	NET-25	4	ETX	0.18	2.5	2.642857143	5.532857143	9.567385052	0.285064735	0.130593999
S	POOL-100	NET-50	4	ETX	0.12	0.625	0.660714286	2.015714286	0.894671606	0.187985659	0.126191601
S	POOL-100	NET-100	4	ETX	0.14	1.25	1.321428571	3.342407991	1.581726214	0.272300884	0.147955819
S	POOL-1000	NET-100	4	ETX	0.51	3.548	4.6568	9.265623918	6.428545487	0.45284636	0.25873771
S	POOL-250	NET-100	4	ETX	0.39	2.7	4.568	8.32328277	4.547478956	0.461977419	0.234973637
S	POOL-2500	NET-100	4	ETX	0.5	5.23	7.54	13.56764227	10.76110588	0.439934491	0.240613271
S	POOL-500	NET-100	4	ETX	0.542981384	2.98	3.5656	7.283914814	8.865594835	0.46022843	0.245846545
S	POOL-750	NET-100	4	ETX	0.31	3.452	4.325	8.690271905	4.10866513	0.420182515	0.233673648
S	POOL-1000	NET-250	4	ETX	0.440945775	14.2135	16.42857143	31.54139244	43.81688931	0.589999101	0.721085481
S	POOL-250	NET-250	4	ETX	0.312	6.25	6.607142857	14.02473849	7.445379063	0.518766253	0.355126397
S	POOL-2500	NET-250	4	ETX	0.31	20.6565	19.658	41.38012929	35.46858361	0.57	0.72
S	POOL-500	NET-250	4	ETX	0.397990398	6.985	7.9898	15.92586498	15.74529642	0.579802423	0.27873748
S	POOL-750	NET-250	4	ETX	0.39	7.658	9.6565	18.26265772	13.82786228	0.58053325	0.411248519
S	POOL-1000	NET-500	4	ETX	0.13	8.265	9.3565	18.51079434	13.70868291	0.618595364	0.497637034
S	POOL-2500	NET-500	4	ETX	0.511312963	21.556	22.6565	44.95410594	59.61724057	0.71	0.46

S	POOL-500	NET-500	4	ETX	0.360988692	13.5	12.21428571	26.6891848	21.88637809	0.597167477	0.40045331
S	POOL-750	NET-500	4	ETX	0.61	13.235	14.9853	29.566289	22.29825708	0.645988087	0.428395266
S	POOL-1000	NET-750	4	ETX	0.46	14.356	15.65568	31.08696801	20.45331547	0.709403723	0.419902747
S	POOL-2500	NET-750	4	ETX	0.575980397	24.5	26.07142857	51.70269836	60.83636088	0.67	0.76
S	POOL-750	NET-750	4	ETX	0.51	19.75	18.82142857	39.84485007	22.67052941	0.675262282	0.426243502
S	POOL-1000	NET-1000	4	ETX	0.342907359	26.8989	25.985	53.87999415	53.09126666	0.809403723	0.403100832
S	POOL-2500	NET-1000	4	ETX	0.46	16.325	17.3268	34.84059555	21.43879348	0.87	0.421497121
S	POOL-2500	NET-2500	4	ETX	0.53	34.3235	35.568	70.81140504	62.29885397	0.91986	0.400455874
S	POOL-100	NET-15	4	SISLOF	0.53	0.812911597	1.240965418	3.38141144	2.107216072	0.272694031	0.18441901
H	POOL-100	NET-15	4	SISLOF	0.79	7.03	16.93041378	26.72455835	1.38474889	0.28028678	0.11258876
S	POOL-100	NET-25	4	SISLOF	0.43	3.996468133	5.704196593	10.48022256	16.81932253	0.429436319	0.259421777
S	POOL-100	NET-50	4	SISLOF	0.17	1.340270872	2.064608308	4.512949564	3.327205892	0.340580602	0.33719027
S	POOL-100	NET-100	4	SISLOF	0.62199684	2.487171556	2.692380657	6.521549053	4.577468824	0.40361037	0.301087797
S	POOL-1000	NET-100	4	SISLOF	0.629328566	9.933149338	12.44814079	23.6006187	29.65068253	0.775530398	0.416607755
S	POOL-250	NET-100	4	SISLOF	0.57	4.606695359	7.432878457	13.33957382	16.30704207	0.563495453	0.535519464
S	POOL-2500	NET-100	4	SISLOF	0.569059564	14.475118	19.10084779	34.49502535	43.26937508	0.692165935	0.59684858
S	POOL-500	NET-100	4	SISLOF	0.61	8.919001436	11.99852058	22.16387505	11.43154815	0.791258472	0.529993423
S	POOL-750	NET-100	4	SISLOF	0.47	5.340824382	6.687288317	13.2181127	27.65135986	0.759755139	0.475463021
S	POOL-1000	NET-250	4	SISLOF	0.892851658	26.56225749	34.34107212	62.69618127	59.88820713	0.705817535	0.695386046
S	POOL-250	NET-250	4	SISLOF	0.41	6.570546308	9.690057996	17.57929771	13.08917853	0.620453897	0.514727772
S	POOL-2500	NET-250	4	SISLOF	0.47	6.735525506	9.293884757	17.30941026	18.26213359	0.697582275	0.41934859
S	POOL-500	NET-250	4	SISLOF	0.51950502	11.53981661	16.70611805	29.36261415	34.58631587	0.630383427	0.729974748
S	POOL-750	NET-250	4	SISLOF	0.42	19.66764335	24.98177319	46.03118487	40.16845871	0.770501409	0.564637243
S	POOL-1000	NET-500	4	SISLOF	0.195915562	21.25263024	28.41356998	50.6779491	54.47355222	0.691280276	0.53279213
S	POOL-2500	NET-500	4	SISLOF	0.848372262	33.21579333	39.37322907	73.88176819	56.15104116	0.686175693	0.57439904
S	POOL-500	NET-500	4	SISLOF	0.4924308	18.31752822	26.23298725	45.75294626	81.88538375	0.675691078	0.517779908
S	POOL-750	NET-500	4	SISLOF	0.73	11.8423038	15.81148391	29.27378771	78.3683822	0.712501958	0.513604032
S	POOL-1000	NET-750	4	SISLOF	0.51	25.62896778	30.10808156	56.98607487	65.38079528	0.763946899	0.665081129
S	POOL-2500	NET-750	4	SISLOF	0.73	27.64955299	36.0442422	65.34007057	49.21904506	0.91	0.534101172
S	POOL-750	NET-750	4	SISLOF	0.651958069	7.668636022	12.24473463	21.40532872	14.42353708	0.679433145	0.429978029
S	POOL-1000	NET-1000	4	SISLOF	0.41	30.54591839	39.61755826	71.26347665	92.31182741	1	0.724192471
S	POOL-2500	NET-1000	4	SISLOF	0.59	23.71827764	30.73462138	55.84668482	30.63922166	1	0.650772132
S	POOL-2500	NET-2500	4	SISLOF	0.655183033	42.78362377	51.96290167	95.87553673	92.07530209	1	0.839583444
S	POOL-100	NET-15	5	OF0	0.48	0.370919032	1.230092556	3.507658747	0.830542549	0.0898972	0.038182941
H	POOL-100	NET-15	5	OF0	0.87	4.91	9.436571269	16.63273959	0.112250383	0.223414457	0.128794
S	POOL-100	NET-25	5	OF0	0.346646137	1.109616558	3.08939488	6.002620688	2.328115937	0.166484198	0.10568968
S	POOL-100	NET-50	5	OF0	0.21	0.66557722	1.541633728	4.052080565	1.003909906	0.258004686	0.120957912
S	POOL-100	NET-100	5	OF0	0.15	1.775208157	3.902601279	7.228983095	2.493645563	0.393788781	0.104821379
S	POOL-1000	NET-100	5	OF0	0.266116104	9.817232496	20.00083775	31.63716041	16.39174947	0.433696099	0.128897777
S	POOL-250	NET-100	5	OF0	0.809316923	5.594446317	11.11053339	18.81563801	7.914582715	0.482510402	0.258260993
S	POOL-2500	NET-100	5	OF0	0.73403602	12.9394411	24.10904008	38.91799487	19.81717564	0.454154225	0.203231415
S	POOL-500	NET-100	5	OF0	0.36834143	6.151508894	13.9728204	21.58217495	13.80415627	0.446467739	0.224878632
S	POOL-750	NET-100	5	OF0	0.801621292	9.580236247	18.30700629	29.85229537	14.19159939	0.450491264	0.0913
S	POOL-1000	NET-250	5	OF0	0.560383075	21.37803799	40.89110647	64.19891047	32.26771759	0.536053376	0.1179702
S	POOL-250	NET-250	5	OF0	0.520478183	10.69623676	20.92203028	33.4750361	17.02864823	0.55383631	0.211010198
S	POOL-2500	NET-250	5	OF0	0.565037318	28.46373156	57.19902933	87.54036942	45.62333389	0.632893532	0.269879084
S	POOL-500	NET-250	5	OF0	0.718021909	11.1656155	25.48318801	38.78443693	17.16062468	0.655109479	0.198225267
S	POOL-750	NET-250	5	OF0	0.456875827	13.42161091	24.60345995	39.75020692	22.0417807	0.714459291	0.220745764
S	POOL-1000	NET-500	5	OF0	0.291200153	28.38315583	25.55557266	55.59209748	39.26661789	0.617709995	0.123092383
S	POOL-2500	NET-500	5	OF0	0.871708556	29.3076288	56.384528	88.01562683	36.88111137	0.608623265	0.23632595
S	POOL-500	NET-500	5	OF0	0.162135856	14.98677511	28.808495	45.23141742	39.26661789	0.66531607	0.228060757
S	POOL-750	NET-500	5	OF0	0.392199627	20.58918453	39.55067504	61.89491633	40.26661789	0.617898026	0.159885563
S	POOL-1000	NET-750	5	OF0	0.177060419	24.95236778	44.5688251	70.90978404	50.06599143	0.684549401	0.18866988
S	POOL-2500	NET-750	5	OF0	0.233479708	31.23446878	60.26503861	93.15141257	55.39029004	0.689528131	0.267141124
S	POOL-750	NET-750	5	OF0	0.334651887	29.08142754	54.71245207	85.38122551	52.78867051	0.702846821	0.324410153
S	POOL-1000	NET-1000	5	OF0	0.116125418	46.8748394	49.78073165	98.18070062	63.37531743	0.752273896	0.365737374
S	POOL-2500	NET-1000	5	OF0	0.613373401	26.12901683	51.535174	79.51031673	42.06874017	0.836766616	0.253362826
S	POOL-2500	NET-2500	5	OF0	0.468478146	37.87744399	68.29916664	107.840708	51.43972984	0.89656	0.41258
S	POOL-100	NET-15	5	ETX	0.63	0.23381621	1.912647976	4.476206456	0.899348018	0.089459268	0.066687848
H	POOL-100	NET-15	5	ETX	0.79	4.06	11.25547286	18.02085766	0.655277534	0.199487574	0.070248273
S	POOL-100	NET-25	5	ETX	0.255421058	2.380809253	4.645386624	8.881303857	4.245158601	0.165256646	0.098469475
S	POOL-100	NET-50	5	ETX	0.36	0.613134418	2.203347386	4.910756103	1.233121158	0.248451267	0.119263115
S	POOL-100	NET-100	5	ETX	0.34	0.985490455	3.61418218	6.478640178	3.194936107	0.429488508	0.144782737
S	POOL-1000	NET-100	5	ETX	0.155729353	5.985878935	10.51495668	18.45031254	10.00367775	0.437214908	0.149844714
S	POOL-250	NET-100	5	ETX	0.541723668	5.272228954	9.476248799	16.65383275	9.187050707	0.447033106	0.143590744
S	POOL-2500	NET-100	5	ETX	0.263783231	8.206560973	15.0442856	24.81019701	17.10438004	0.459563243	0.231452638
S	POOL-500	NET-100	5	ETX	0.28426014	4.500640794	8.592633172	14.51574502	9.204597802	0.40938942	0.158828151
S	POOL-750	NET-100	5	ETX	0.357186392	5.139790717	11.83430345	18.8610715	14.30026817	0.443357285	0.060041733
S	POOL-1000	NET-250	5	ETX	0.689408996	18.09035432	33.31676546	53.53424388	26.67556277	0.667393832	0.280219888
S	POOL-250	NET-250	5	ETX	0.176348636	7.702871154	16.10687763	25.42122957	16.86863114	0.598613133	0.282736083
S	POOL-2500	NET-250	5	ETX	0.676515785	20.88216716	44.37054265	67.37867301	33.88055894	0.605784809	0.150458389
S	POOL-500	NET-250	5	ETX	0.701149854	8.533017581	17.66530898	28.47944006	14.70895961	0.700097082	0.173643714
S	POOL-750	NET-250	5	ETX	0.208982017	11.13550934	19.97665729	32.80135167	18.41844799	0.660783377	0.20570603
S	POOL-1000	NET-500	5	ETX	0.38	22.20389767	29.67394578	53.70840118	29.68445284	0.634361009	0.208815951

S	POOL-2500	NET-500	5	ETX	0.478444043	24.81958133	47.92239556	74.96188881	59.62887634	0.698873895	0.260176832
S	POOL-500	NET-500	5	ETX	0.186343343	14.25066215	31.05453991	46.85374046	36.28659117	0.674714359	0.179847724
S	POOL-750	NET-500	5	ETX	0.850023538	15.86804059	31.04996247	49.31457883	27.65452858	0.616115346	0.162347954
S	POOL-1000	NET-750	5	ETX	0.422298226	35.05958229	39.58729529	76.60160765	38.18781877	0.701058408	0.234497076
S	POOL-2500	NET-750	5	ETX	0.477472032	27.15031772	53.16828679	82.43579261	44.19938932	0.71063999	0.290343088
S	POOL-750	NET-750	5	ETX	0.779745388	20.37951945	43.65981961	66.27148768	34.95313554	0.715943847	0.276281076
S	POOL-1000	NET-1000	5	ETX	0.216195087	30.74552559	63.64575698	96.18781724	43.37531743	0.847787457	0.313064941
S	POOL-2500	NET-1000	5	ETX	0.220551871	19.60534332	38.37823815	59.78503865	52.64261069	0.868361743	0.337374412
S	POOL-2500	NET-2500	5	ETX	0.777419473	40.74293038	64.30455331	107.1698479	49.49550181	0.93748	0.4922645
S	POOL-100	NET-15	5	SISLOF	0.69	0.875489698	3.387870254	6.807916363	6.720741178	0.308376353	0.288704325
H	POOL-100	NET-15	5	SISLOF	0.98	5.46	7.807938316	16.26244351	7.36	0.145051908	0.113265
S	POOL-100	NET-25	5	SISLOF	0.53	2.722732706	6.398550471	11.52927088	5.89187311	0.462928569	0.406807646
S	POOL-100	NET-50	5	SISLOF	0.49	1.510455936	2.9201545	6.931995963	20.75146111	0.486408929	0.314336013
S	POOL-100	NET-100	5	SISLOF	0.96822832	1.862581305	6.502302954	11.03311258	4.460917858	0.462783157	0.415151797
S	POOL-1000	NET-100	5	SISLOF	0.858521524	12.03923606	31.3598665	46.17806806	27.48578268	0.775784053	0.462370472
S	POOL-250	NET-100	5	SISLOF	0.751681603	7.214550496	15.66186479	25.08086039	28.43750969	0.695345226	0.419911186
S	POOL-2500	NET-100	5	SISLOF	0.683199407	18.14361554	43.83074071	64.09779325	55.16495895	0.736667939	0.705008874
S	POOL-500	NET-100	5	SISLOF	0.78	10.86281605	26.76587335	40.33668451	34.47794616	0.808084409	0.484624391
S	POOL-750	NET-100	5	SISLOF	0.94	7.242162785	16.45107122	26.7880207	74.83590331	0.648099258	0.52558932
S	POOL-1000	NET-250	5	SISLOF	0.987580553	29.65017789	71.20902914	103.4195678	64.63838768	0.727446787	0.681984187
S	POOL-250	NET-250	5	SISLOF	0.655125836	9.217464659	22.22388708	33.94570266	20.90027751	0.587963718	0.508916202
S	POOL-2500	NET-250	5	SISLOF	0.524362102	9.442937987	22.08390398	33.89943871	22.96875772	0.702821733	0.370547761
S	POOL-500	NET-250	5	SISLOF	0.79	15.42399511	32.38677125	50.3655114	85.99517701	0.756283378	0.682600484
S	POOL-750	NET-250	5	SISLOF	0.585643312	21.88840657	51.09297211	75.22298019	54.39870052	0.743885906	0.601709189
S	POOL-1000	NET-500	5	SISLOF	0.43	35.55549275	61.03800248	98.68303047	95.66965053	0.643190379	0.547002534
S	POOL-2500	NET-500	5	SISLOF	0.89	45.76147013	81.9832118	130.4894705	146.0225077	0.717751088	0.59122062
S	POOL-500	NET-500	5	SISLOF	0.834482996	23.48559887	53.86860779	79.76392983	76.6093212	0.601043865	0.541925494
S	POOL-750	NET-500	5	SISLOF	0.844759247	30.16891313	39.78000329	72.52655676	64.9093338	0.666854944	0.53684359
S	POOL-1000	NET-750	5	SISLOF	0.636087941	29.21550261	72.81794004	104.2915007	118.912053	0.801341657	0.560551658
S	POOL-2500	NET-750	5	SISLOF	0.865179093	32.04726432	75.54059059	110.3032284	96.45904191	0.83	0.608432732
S	POOL-750	NET-750	5	SISLOF	0.8848706	13.13415478	28.8747202	44.64365867	25.69690123	0.668020435	0.483799163
S	POOL-1000	NET-1000	5	SISLOF	0.49	34.66621593	81.32681597	118.2481381	89.00165336	0.861340077	0.787275222
S	POOL-2500	NET-1000	5	SISLOF	0.76	27.79438932	64.3965801	94.60692463	88.77035594	1	0.670130386
S	POOL-2500	NET-2500	5	SISLOF	0.770207218	46.93462688	95.68726085	145.1742553	98.87378754	1	0.864945261

Appendix C

Fixed Network Experiment

C.1 Fixed Network Experiment 100 keys Pool

Pool			
01101110	11011000	10100001	10100111
11101001	01011110	11001110	11010101
00100010	11000111	10100100	11001001
10111001	00101000	00111000	01011000
00110110	00001101	11100000	01101101
10011001	11100010	10000000	11101000
10010110	10111010	10110110	10010000
00010001	10011011	01001110	11000101
11011100	10010101	01001001	00101010
01111111	01100000	10000101	01100100
10011100	00110011	01011001	01010100
11101110	00000101	11001011	01011111
00011000	00010110	10001011	00001000
00101101	00011010	00101110	00110101
10110011	01111110	00110000	00011011
00111001	10010010	01001111	10100010
01000110	01110011	00101001	10101110
11010110	10000010	11011010	00100001
10000011	10011111	10001111	01011101
10101001	01010000	00000111	01000100
00010101	10111100	01000010	10100000
00011111	11111100	11000001	11101010
10101101	11001000	00111010	01011011
11111010	11001101	10100011	11001100
01110000	01010111	00001001	01101101

Table 3.1: 100 keys Controlled experiment pool

C.2 Fixed Network Experiment Nodes rings

Node 1	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
10101110	10011100	10100011	10000010	01100100	11010101	11101010	00111001
11101110	11000101	01101101	10010101	10010000	00001000	11001011	00011011
01110000	01010100	10010010	11011000	01100000	01000010	10000010	01000100
11101001	10101110	01110000	00001101	00011000	01100100	11100010	10010000
11000111	10011111	10010000	11000101	11100010	01111111	01001111	11001011
11000101	01111111	10010010	11111010	00111010	00111010	11011000	10101101
00101010	00111000	10000010	10101101	11101110	11100000	01010111	11001000
11001001	11100000	11010110	00110011	01011110	00010001	11001000	10000010
Node 9	**Node 10**	**Node 11**	**Node 12**	**Node 13**	**Node 14**	**Node 15**	
01011101	00011111	10100011	10001011	11010101	11001011	10010110	
11001001	01001001	10100100	10110110	10011100	01110000	01100100	
00101001	00101001	10011100	00110011	11010101	11001011	10100011	
10011111	01101101	10111001	01000110	01111111	01000100	00010110	
10110110	10000010	01111111	11011000	00001101	11001001	11010101	
00000111	10101110	01011110	11111100	01011101	10100011	11001011	
11011010	10100011	11111010	01101101	01100100	10110011	00101000	
00111001	00011000	00011000	10111001	01100100	11000111	10010101	

Table 3.2: Rings for 15 nodes in the controlled experiment

C.3 Fixed Network Experiment shared keys

Shared keys		
$node1(ring) \cap node2(ring)$,	$node3(ring) \cap node14(ring)$,	$node7(ring) \cap node10(ring)$,
$node1(ring) \cap node3(ring)$,	$node3(ring) \cap node15(ring)$,	$node7(ring) \cap node12(ring)$,
$node1(ring) \cap node4(ring)$,	$node4(ring) \cap node7(ring)$,	$node7(ring) \cap node14(ring)$,
$node1(ring) \cap node5(ring)$,	$node4(ring) \cap node8(ring)$,	$node7(ring) \cap node15(ring)$,
$node1(ring) \cap node9(ring)$,	$node4(ring) \cap node10(ring)$,	$node8(ring) \cap node9(ring)$,
$node1(ring) \cap node10(ring)$,	$node4(ring) \cap node11(ring)$,	$node8(ring) \cap node10(ring)$,
$node1(ring) \cap node14(ring)$,	$node4(ring) \cap node12(ring)$,	$node8(ring) \cap node14(ring)$,
$node2(ring) \cap node4(ring)$,	$node4(ring) \cap node13(ring)$,	$node8(ring) \cap node15(ring)$,
$node2(ring) \cap node6(ring)$,	$node4(ring) \cap node15(ring)$,	$node9(ring) \cap node10(ring)$,
$node2(ring) \cap node9(ring)$,	$node5(ring) \cap node6(ring)$,	$node9(ring) \cap node12(ring)$,
$node1(ring) \cap node10(ring)$,	$node5(ring) \cap node7(ring)$,	$node9(ring) \cap node13(ring)$,
$node1(ring) \cap node11(ring)$,	$node5(ring) \cap node8(ring)$,	$node9(ring) \cap node14(ring)$,
$node2(ring) \cap node13(ring)$,	$node5(ring) \cap node10(ring)$,	$node10(ring) \cap node11(ring)$,
$node3(ring) \cap node4(ring)$,	$node5(ring) \cap node11(ring)$,	$node10(ring) \cap node12(ring)$,
$node3(ring) \cap node5(ring)$,	$node5(ring) \cap node13(ring)$,	$node10(ring) \cap node14(ring)$,
$node3(ring) \cap node7(ring)$,	$node5(ring) \cap node15(ring)$,	$node10(ring) \cap node15(ring)$,
$node3(ring) \cap node8(ring)$,	$node6(ring) \cap node11(ring)$,	$node11(ring) \cap node12(ring)$,
$node3(ring) \cap node10(ring)$,	$node6(ring) \cap node13(ring)$,	$node11(ring) \cap node13(ring)$,
$node3(ring) \cap node11(ring)$,	$node6(ring) \cap node15(ring)$,	$node11(ring) \cap node14(ring)$,
$node3(ring) \cap node12(ring)$,	$node7(ring) \cap node8(ring)$,	

Table 3.3: Shared keys between nodes in the controlled experiment

Appendix D

Published Research

2016 IEEE International Conference on Cloud Engineering Workshop

Securing the Internet of Things Devices Using Pre-Distributed Keys

Ayman El Hajjar
Department of Computer Science
Birkbeck, University of London
Email: ayman@dcs.bbk.ac.uk
Supervised by Professor George Roussos Dept of Computer science
& Dr Maura Paterson Dept. of Economics, Mathematics and statistics

Abstract—**The paper outlines the state of the art, problems and challenges in the Internet of things (IoT) security. It investigates how the key pre-distribution algorithm of Eschenauer and Giglor designed for Distributed Sensor Networks(DSN) performs when applied on the IoT for 6LoWPAN networks. A simulation that uses the Contiki Operating System was developed in order to explore the performance of the algorithm on those devices. After an explanation of the research methodology and the details of the experiment conducted, we present the results from the experiment in comparison with the results obtained by Eschenauer & Giglor.**

I. Introduction and Motivation

The Internet of Things refers to a world-wide network of interconnected heterogeneous objects (sensors, actuators, smart devices, smart objects, RFID, embedded computers and so on) uniquely addressable based on standard communication protocols [1]. In a common Wireless Sensor Network (WSN), each node plays an important role to ensure data confidentiality, integrity, availability and authentication. For those nodes to be attacked, it requires a physical presence near the targeted node in order to attack it. The Internet of Things interconnects WSN networks to the Internet and thus there is no need for location proximity and attacker would be able to attack any WSN node from the Internet.

For this reason, authentication between devices communicating in the IoT network became a necessity. This includes securing messages transmitted in the Routing Protocol for lossy Networks (RPL) as the security was not part of the protocol standard. Threats due to authentication failure is a main issue for motes joining the RPL Routing table as discussed in the IETF Routing Over Low Power and Lossy networks (ROLL) security threats draft [2]. A suggestion to use keys pre-distribution was made in the protocol draft. This suggestion did not specify which key pre-distribution protocol to use.

This paper suggests the use of the key-pre distribution algorithm proposed by Eschenauer & Giglor in the context of the IoT using 6LoWPAN adaption layer protocols.

II. Background to Research topic

6LoWPAN: Low Wireless Personal Area Networks are simple low cost communication networks that allow wireless connectivity in devices with limited power and relaxed throughput requirements. The 6LoWPAN adaptation layer concept originated from the idea that Internet Protocols could and should be applied even to the smallest devices, and that low-power devices with limited processing capabilities should be able to participate in the IoT [3]. Internet Protocol version 4 will not be able to accomodate the large number of LR-WPAN devices that is expected to be deployed in the IoT and thus IPv6 will be used for addressing of the IoT devcices. [4] Some are potentially left unattended or hard to reach and in harsh conditions. Any protocol used on those networks should take into consideration this unreliable nature of communication [5].

RPL: Routing in Low Power and Lossy networks (LLN) should be able to self manage and to self heal without requiring manual intervention. Routing Protocol for Low-Power and Lossy Networks (RPL)

978-1-5090-3684-4/16 $31.00 © 2016 IEEE
DOI 10.1109/IC2EW.2016.22

is a distance vector IPv6 routing protocol designed for Low-Power and Lossy Networks (LLNs). RPL constructs a Directed Acyclic Graph (DAG) that attempts to minimize path costs to the DAG root according to a set of metrics and objective functions [5]. RPL draft includes two security modes, one called preinstalled, where motes joining an RPL instance have preinstalled keys that enable them to process and generate secured RPL message and another mode that is called authenticate. In authenticated mode motes have preinstalled keys as in preinstalled mode, but the preinstalled key may only be used to join a RPL instance as a leaf [6].

Keys Pre-Distribution for Distriuted Sensor Networs DSN: Traditional key exchange and key pre distribution protocols based on infrastructure using trusted third parties are impractical for large scale distributed sensor networks. A key management scheme for distributed sensor networks DSN proposed in [7] requires memory storage for only a few tens to a couple of hundred keys, and yet has similar security and superior operational properties when compared to those of the pair wise private key sharing scheme. This scheme relies on probabilistic key sharing among the motes and uses a simple shared key discovery protocol for key distribution. First and prior to DSN deployment, a ring of keys is distributed to each sensor mote, each key ring consisting of randomly chosen k keys from a large pool of P keys which is generated offline. Even if two motes do not share a key the pair of motes can use the path of an existing pair wise path to exchange keys and establish a direct link. This ensures that even when only the probability of the links between motes to share a key is 0.5, a fully secure communication network can be guaranteed 99.999% as long as multilink paths of shared keys exist among neighbours [7].

III. Problem Definition and challenges

Providing security in IPv6/RPL connected 6LoWPANs is challenging because the devices are connected to the untrusted Internet and are resources constrained and the communication links are lossy [8]. The interest of this paper lies in Protocol Translation and End to End Security challenge [9]. The keys pre distribution algorithm suggested by [7] for Distributed Sensor Networks (DSN) was implemented for wireless sensors differs from a network of 6LoWPAN devices using RPL. This presents challenges such as in a DSN network if a mote does not share a key with one of its neighbours, it uses multi-link path to communicate with it, in contrast with IoT network where nodes are using RPL and each mote can communicate only with the mote that it form a leaf with. The limitations and constraints of the IoT devices also present another challenge in term of memory and processing power which mean a limitation in the size of keys, IDs and Rings.

IV. Proposed Approach

We propose to implement the Keys Pre Distribution for Distributed Sensor Networks DSN discussed in [7] on IoT devices network using RPL routing protocol. The key pre-distribution algorithm for DSN to the best of our knowledge was never tested on IoT devices using RPL routing protocol. We developed a simulation experiment to test our algorithm implementation.

V. Research & Experiment Methodology

The simulation was developed on the Contiki Operating System [11]. It uses many applications and tools designed specifically for low power lossy Networks and IPv6 devices such as Cooja [13] and Tunslip6 [11] The simulation experiment is looking at the performance of the key pre distribution algorithm proposed in [7] in the context of RPL.

The simulation experiment is looking specifically to explore the percentage of leaves in the RPL routing table that share a key. The sizes of bothe values are obtained using the same formulaes used in [7]. the key Ring and the Pool ranges from 8 keys in a key Ring when the Pool has 100 keys to 41 keys and in a key Ring when the Pool has 2500 keys. Those values are obtained using the same formulaes used in [7].

keys in the pool were generated and distributed thatto key Rings randomly using different Random techniques. Keys in the pool were generated using Blum Blum shub random number generator [14]. IDs were generated using Random library from C libary. Keys and IDs were distributed to differents Rings using knuth shuffle random algorithm [15].

VI. preliminary findings

The results of the simulation experiments shows that out of each pool used, a big proportion of the leaves in the routing table shared a key as shown in figure 1 below[1]. For example, in figure 1 below, when the Pool contained a 1000 keys and the network was of 1000 motes, the percentage of motes in the DODAG that has a shared key was 54.01%. From the results obtained, it is clear that the internet of things devices when simulated achieve an average probability close to the 0.5% claimed. However this probability is not enough to achieve full connectivity of the network when using the RPL routing protocol since only a propotion of the leaves in the RPL table has a shared key and can communicate securly. However this leaves the remainder of the routing table leaves with unsecured links.

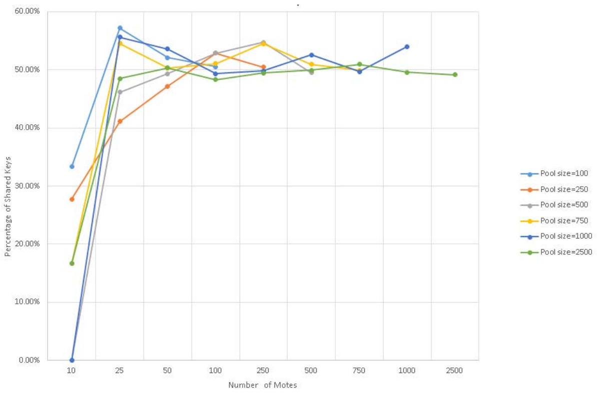


Fig. 1. Number of motes Vs Percentage of shared keys for various pools size

VII. Conclusion and Future Work

This paper investigated the performance of the key pre-distributed algorithm for distributed sensor networks on the IoT devices. The results obtained shows that the keys pre distribution algorithm when implemented on the IoT network using RPL does not achieve full secure connectivity in contrast with the DSN network in [7] since not all the RPL leaves are secured and thus not all motes in the RPL routing table are able to communicate.

The next step in this research will be to explore alternatives for solutions regarding leaves in the RPL routing table that do not share a key. A promising solution is to look at the Reactive Discovery of Point to Point routes in Low Power and Lossy Networks. [16].

[1] Percentage of shared keys for 10 or 25 motes in the network is low as motes are unable to communicate with each other

References

[1] Giancarlo Fortino & Paolo Trunfio, Internet of Things Based on Smart Objects: Technology, Middleware and Applications, Springer Publishing, 2014

[2] Rsao, et al., A Security Threat Analysis for Routing Protocol for low power and lossy Networks (RPL), Internet Engineering Task Force (IETF),draft-ietf-roll-security-threats-10, September 2014 `https://tools.ietf.org/html/draft-ietf-roll-security-threats-10`

[3] IEEE 802.15.4 Low Rate Wireless Personal Area Networks (LR-WPANs), IEEE standard for local and metropolitan area networks, USA, September 2011

[4] Zach Shelby & Carsten Bormann 6LoWPAN:The wireless embedded Internet-Part 1:Why 6LoWPAN?" EE Times,May 2011

[5] Hui & Vasseur, The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams, IETF, RFC 6553 March 2012 `https://tools.ietf.org/html/rfc6553`

[6] Winter, et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, IETF, RFC 6550, March 2012 `https://tools.ietf.org/html/rfc6550`

[7] Laurent Eschenauer & Virgil D. Gligor, A key Managemenent Scheme for Distributed Sensor Networks,Proceedings of the 9th ACM conference on Computer and Communication security USA, 2002

[8] Linus Wallgren, Shahid Raza &nd Thiemo Voigt, Routing Attacks and Countermeasures in the RPL-Based Internet of Things, Hindawi Publishing Corporation International Journal of Distributed Sensor Networks Volume 2013, Article ID 794326, 11 pages, June 2013. `http://dx.doi.org/10.1155/2013/794326`

[9] Alfred J. Menezes, Paul C. Van Oorschot and scott A. Vanstone, Handbook of Applied Cryptography, fifth edition, CRC press, August 2001. `http://cacr.uwaterloo.ca/hac/`

[10] Gradia-Morchon, et.al, Security consideration in the IP based Internet of Things, IETF, Internet Draft, March 2012

[11] Contiki Operating system `http://contiki-os.org`.

[12] Frederik Ostrelind, A sensor Network Siumulator for the Contiki OS, February 2006 `http://soda.swedish-ict.se/2296/1/SICS-T--2006-05--SE.pdf`.

[13] Frederik Ostrelind, A sensor Network Siumulator for the Contiki OS, February 2006 `http://soda.swedish-ict.se/2296/1/SICS-T--2006-05--SE.pdf`.

[14] Lenore Blum, Manuel Blum, Michael Shub, Comparison of two Pseudo -Random number generators,plenum, 1982

[15] Donald E Knuth, The art of computer programming, Volume 2, Seminumerical algorithms, Adison Welsey Reading, 1969

[16] Goyal, et al. Reactive Discovery of Point to Point Routes in Low Power and Lossy Networks, Internet Engineering Task Force, draft-etf-roll-p2p-rpl-07, January 2012 `http://tools.ietf.org/html/draft-ietf-roll-p2p-rpl-07`

On the Performance of Key Pre-distribution for RPL-based IoT Networks

Ayman El Hajjar[1], George Roussos[1] and Maura Paterson[2]
[1]Department of Computer Science and Information Systems
[2]Department of Economics, Mathematics and Statistics
Birkbeck University of London
{a.elhajjar, g.roussos, m.paterson}@bbk.ac.uk

Abstract

A core ingredient of the the *Internet of Things (IoT)* is the use of deeply embedded resource constrained devices, often connected to the Internet over Low Power and Lossy Networks. These constraints compounded by the need for unsupervised operation within an untrusted environment create considerable challenges for the secure operation of these systems. In this paper, we propose a novel method to secure an edge IoT network using the concept of key pre-distribution proposed by Eschenauer and Gligor in the context of distributed sensor networks. First, we investigate the performance of the unmodified algorithm in the Internet of Things setting and then analyse the results with a view to determine its performance and thus its suitability in this context. Specifically, we investigate how ring size influences performance in order to determine the required ring size that guarantees full connectivity of the network. We then proceed to propose a novel *RPL objective function* and associated metrics that ensure that any node that joins the network can establish secure communication with Internet destinations.

1 Introduction

In recent years, with the development of wireless sensor networks, the Internet of Things (IoT) became a reality. This presents many challenges that also did not exist before because of the nature of the IoT. Since the IoT is a collection of heterogeneous networks, it involves not only the same security problems with sensor network, but also more particular ones, such as privacy protection problem, heterogeneous network authentication and access control problems, information storage and management [1].

The research into the IoT security is far more complicated then that of the Internet security in general. Conventional security protocols for the Internet as we know are not suitable for the Internet of Things. Devices in the IoT are different in terms of computation capabilities, memory limitation, processing power and physical limitation (i.e., installed in rural area and unattended). Thus factors such as reliability, scalability, modularity, interoperability, interface and QoS can be hard to achieve [2].

Security of the Internet of Things is at the centre of research. The impact of security breaches on humans in an IoT device is much greater than in conventional networks. For example, a breach of a device monitoring the CO_2 level in a room can lead to physical harm to a human being if this device is compromised and is sending data that are not accurate. Thus authentication and authorization are key to ensuring that only authenticated devices (those that share a suitable key) can join the network. The main challenge, when it comes to authentication of various IoT devices, is the design of key storage and distribution mechanisms, because of the nature of the IoT devices and their network architecture [3].

Given the limitation that IoT devices (sensors and actuators) are constrained in term of computational power and storage memory, several of the conventional security methods are not suitable for use.

The purpose of this paper is to investigate the performance of Laurent Eschenauer and Virgil D. Gligor's Algorithm [4] for Distributed Sensor Networks (DSN) in the context of IPv6 Low Power and Lossy Networks (6LoWPAN) Devices for the Internet of Things (IoT). We provide an analysis of the performance of the algorithm when applied in the DSN and IoT context. We also show the ring size needed to guarantee full network connectivity. We then propose a modification of the routing protocol for Low power and Lossy Networks (RPL) Objective function (OF) in order for the key pre-distribution algorithm to achieve a full network connectivity in the context of the IoT.

Section 2 provides an introduction to the Internet of Things, the 6LoWPAN network protocol, the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) and several solutions that attempts to secure the Internet of Things. Section 3 presents the key pre-distribution algorithm by Eschenauer and Gligor in [4]. In section 4, we present the experiment methodology and design that we carried in order to first validate the results of [4] and second to determine whether those results are applicable in the context of the IoT. In section 5 we provide an overview of the future work that will be carried on to enable key pre-distribution algorithm to become a suitable solution for the IoT. Finally, we present our main conclusions in Section 7.

2 Understanding the Problem: Literature Review

Distributed Sensor Networks (DSN) include a large array of sensor nodes that are usually battery powered, have limited computational capabilities and memory. Nodes in a DSN network, collect data and make it available for processing to application components of the network and control nodes. The scale of a DSN network is quite large (tens of thousands). The Internet of Things (IoT) network is a collection of sensor networks (Wireless and Distributed) that share the same characteristics as Distributed Sensor Networks.

2.1 Internet of Things and 6LoWPAN

Internet of Things is a simple low cost communication network that allows wireless connectivity in applications with limited power and relaxed throughput requirements [5]. 6LoWPAN concept originated from the idea that "the Internet Protocol could and should be applied even to the smallest devices" and that low-power devices with limited processing capabilities should be able to participate in the Internet of Things [6].

Internet protocols has always been considered too heavy for sensor networks and thus the 6LoWPAN protocol stacks were created. The need for an IP based sensor network made many researchers attempt to adapt existing Internet standards to the creation of interoperable protocols and the development of supporting mechanisms for composable services [7]. Not surprisingly, one of these challenges is security because of the distinct features of sensor networks such as the capabilities of the nodes. In section 2.3, we will review the various attempts to create new security protocols for sensor networks and the IoT or to adapt existing protocols in the context of the IoT.

Given those limitations, another problem arises with IP for the 6LoWPAN network stacks that is relevant to this paper, the topology of the network. Various topologies should be supported by 6LoWPAN networks including mesh and star. Routing for Low Power and Lossy network (RPL) as described in [8], is a routing protocol for 6LoWPAN networks that can solve this problem.

2.2 Routing for Low Power and Lossy Networks RPL

The Routing Protocol for Low-Power and Lossy Networks (RPL) is a distance vector IPv6 routing protocol designed for LLN networks. RPL is designed for networks which comprise of thousands of nodes where the majority of the nodes have very constrained energy and/or channel capacity. To conserve precious resources, a routing protocol must generate control traffic sparingly. However, this is at odds with the need to quickly propagate any new routing information to resolve routing inconsistencies quickly.

RPL organises its topology in a Directed Acyclic Graph (DAG). An RPL DAG must have at least one RPL root and a Destination Oriented DAG (DODAG) is constructed for each root. The root acts as a sink for the topology by storing all routes to all nodes in the DODAG in the routing table. The root may also act as a border router for the DODAG to allow nodes that belong to different DODAGs to communicate [8].

RPL supports three security modes: unsecured, preinstalled and authenticated. Unsecured refers to the security mechanism that is provided in lower layers such as link layer security. Preinstalled and authenticated modes require the use of preinstalled shared keys on all nodes prior to deploying the nodes. Both modes provide security procedures and mechanisms at the conceptual level and are concerned with authentication, access control, data confidentiality, data integrity and non repudiation. This study focuses on the preinstalled mode as a method of securing message transmission between nodes in an RPL DAG instance.

Authentication in the preinstalled mode involves the mutual authentication of the routing peers prior to exchanging route information (i.e. peer authentication) as well as ensuring that the source of the route data is from the peer (i.e. data origin authentication) [9]. The limitation of the preinstalled mode in its common form, is that it is assumed that a node wishing to join a secured network is pre-configured with a shared key for all neighbours and the RPL root. This means that once this shared key is compromised, all network leaves in the RPL DODAG are compromised.

2.3 Security for the Internet of Things proposed solutions

Providing key management for confidentiality and group level authentication in a sensor network is difficult due to the ad hoc nature and limited resources of the distributed sensor network environment. The main challenge in public key algorithms when using in the context of Internet of Things, similarly to sensor networks, is the energy consumption of exchanging public key certificates [10].

Key management protocols can be divided into three categories. Arbitrated keying protocols, Self Enforcing protocols and Pre-Deployed Keying protocols.

Arbitrated keying protocols requires a trusted server such as the use of [11]. They are not suitable for use in the context of the IoT because of the limited energy, communication bandwidth and computational capacities of sensor nodes in an IoT network. The Otway-Rees protocol in [12] is applied in the context of the IoT for one-way authentication; symmetric cryptography with AES is used for encryption. The drawback in one way authentication is that it leaves the network vulnerable to man-in-the-middle attacks.

Self Enforcing protocols such as Pairwise Asymmetric Keying are based on the Diffie-Hellman key agreement protocol. A proposed solution to use a lightweight DTLS based keying mechanism to secure IoT was suggested in [13]. Although this solution proved to provide a lighter and robust security protocol using pairwise key establishment between nodes, the number of message transfers to establish the secure connection in [13] still introduced a large communication overhead. Pre-deployed keys into nodes prior to deployment in a network offers energy efficient solution to providing confidentiality and group level authentication keys [10].

In the next section we investigate the use of the key management scheme for Distributed Sensor Networks proposed by Eschenauer and Gligor in [4] in the context of the Internet of Things.

3 Key Pre Distribution as a solution for Securing IoT

Offline key pre-distribution algorithm for DSN by Eschenauer and Gligor [4] describes the method by which keys are distributed to nodes in the network.

This key pre-distribution mechanism ensures that for each direct link between any two nodes in the network, the probability of those two sharing at least a key is 0.5. The authors of [4] concluded that the size of key rings and identifier rings $RING$ does not need to be large in order for a network to guarantee full connectivity and only 50% of those pair of nodes need to have a shared key.

At first, a large pool P of keys K and identifiers I is generated. Each key K in the pool is randomly represented by one of the identifiers I. A certain number of identifiers K and their respective keys K are picked from the pool P randomly and loaded into the memory of the node. This will form the key ring and the identifier ring. This step will be repeated for each node that wishes to join the network.

Now that each node in the network has an identifier ring and a key ring loaded into its memory, nodes can begin the phase of selecting a secure route to any other nodes. Each node broadcast its identifier ring to all neighbouring nodes (neighbouring nodes are the nodes that are within it is transmission range). Each neighbouring node compares the identifier ring it received with its own identifier ring. If the node find a shared identifier between the two identifier rings, it sends a message to the origin node with the shared identifier. Nodes that have a shared identifier can establish a secure direct link by using the key that corresponds to the shared identifier. Nodes that do not share an identifier with the origin node will attempt to create a link with it through other nodes (indirect links by hops).

An example in [4] showed that when a pool contained 100,000 keys, full network connectivity was achieved with only 75 keys in the rings. This is due to the fact that routing in Distributed Sensor Networks (DSN) allows multi hops and indirect hop communication between nodes, thus nodes that do no share an identifier can use another node that it shares an identifier with as an indirect link to reach it.

This paper is attempting to evaluate the performance of this algorithm in the context of the IoT environment when using RPL.

4 Experiment Design and Setup

The experiment was simulated on the Contiki Operating System [14] using Cooja nodes simulator [15]. A C program was coded to generate keys pool, IDs pool, Key rings, ID rings [1]. The simulation file was composed of N nodes

[1] Keys & identifiers were generated randomly using Blum Blum Schub generator. Each node will then choose a set of Keys & identifiers for its key ring and identifier ring randomly using Knuth Shuffle algorithm.

and one border router [2]. A script was written in order for the simulation to stop running only when all possible routes were computed and no more routes exist. This was essential to ensure that the routing table we obtain at the end of each simulation is the optimum one for our setting. Finally, a Perl program was coded to analyse logs generated by individual nodes after simulation in order to determine if nodes were able to establish a secure link.

4.1 Experiment parameters

The parameters selected for the simulation experiments aim to approximately match the characteristics of a recent innovative deployment of IoT technology at the campus of the University of Liverpool in the UK, where 650 students were able to employ a smartphone app to access discounts or coupons in stores or cafeterias, as well as for wayfinding and alerting. Specifically, the overall area of 250x250 meters which is a typical area size of a medium sized university campus. Number of users (Network size) is based on an average number of wifi usage at Birbkeck campus during a day which is 2394 users [17]. The main difference between out simulations and the use case we use for motivation is the wireless technology used which was Bluetooth Low Energy (BLE) while in the simulations we use Zigbee.

Parameters related to the environment (control parameters) of the simulation were defined in the experiment configuration. We assumed that the transmitting range for each node is 50 meters (this is the common transmitting range for 6LoWPAN low power devices). We also used the key length *klength* of 64 bits and the ID length *ilength* of 32 bits. Those two sizes were chosen as they are enough, given the number of nodes we simulated in the experiment. The number of bits in ID was chosen to be smaller because of memory constraints in the Internet of things devices. The other reason is that exchanging IDs is not revealing anything as there is no connection between keys and IDs is exchanged. Anyone trying to intercept the messages will not be able to make the connection between the identifer exchanged and the key it represents.

We carried out the experiment simulations with three different parameters (independent parameters) changing. The Pool size P of keys is the first parameter. Two pools are being generated in each simulation, one for keys, the other for IDs. Both have the same size. The pool size is an important factor that will have a huge impact on the probability of shared keys between motes. The pools size we run simulations for are: 100, 250, 500,750,1,000 and 2,500 motes. The second parameter is the network size N. The third parameter is the ring size RS It was computed using Stirling equation as per [4]. Those independent variables are shown below and in table 1. For each pool size (P), keys and identifers are generated once for all networks size. To ensure the accuracy of experiment simulations, each simulation will run 5 times. The largest and smallest results were discarded and the average of the remaining three runs is used. The outputs of

[2] A border router is also the root of the RPL DODAG and it will store the routing table of the simulation (acting as a sink).

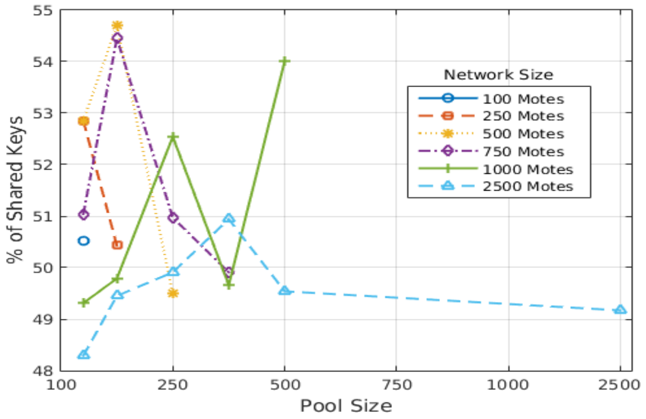


Figure 1: Number of nodes Vs Percentage of shared keys for various pools size

each of those experiments are the Number of DAGs $DAGs$ in the routing table and the Number of Shared Keys NSK between nodes that formed a DAG.

Table 1: Independent Variables

Pool size (P)	Ring size (RS)	Network size (N)					
100	8	100					
250	13	100	250				
500	18	100	250	500			
750	22	100	250	500	750		
1,000	25	100	250	500	750	1,000	
2,500	41	100	250	500	750	1,000	2,500

4.2 Experiment Results and Analysis

Fig. 1 shows the percentage of shared keys for various pools size when changing the density of nodes in the network in a small environment of 250 by 250. As we can see from fig. 1, the result of percentage of shared keys in the $DAGs$ becomes consistent around 50%. If the network simulated is a Distributed Sensor Network, a 50% of links between various nodes in the DSN network sharing a key is enough to guarantee full connectivity of the network. In a DSN network, nodes that do not share a key can use a neighbouring node as an indirect link as long as the link is secure. This will mean that it will take the connection between two nodes two hops rather than a direct link but both of them will be secure. However this network is an IoT network, therefore nodes that do not share a key in the routing table will be discarded. Point to Point links in RPL routing is not allowed therefore an alternative multihop secure link can not exist.

Fig. 2 represents the ring size vs the percentage of shared keys in the DAG for various Network size. In this graph, it is very clear that the percentage of shared key $\%NSK$ is hovering around the 50%. We can also validate from fig. 2 that the size of the ring calculation used in [4] generated a 50% shared keys between nodes in the DSN network. The percentage of $DAGs$ that contains a shared key can also be validated for IoT as 50% of the RPL routing table leaves had a common key ($\%NSK$) in the ring.

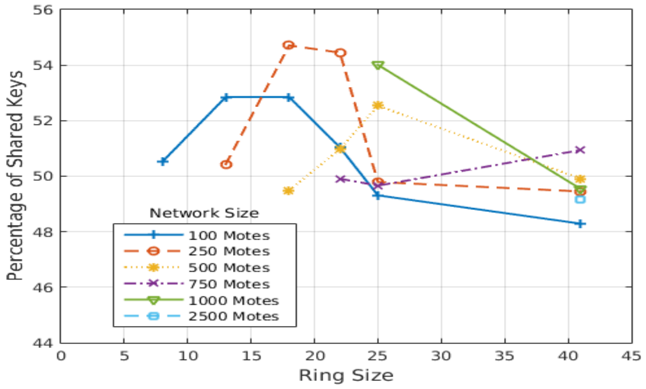
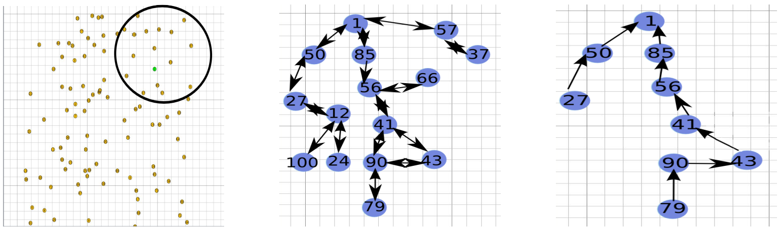


Figure 2: Ring size Vs Percentage of shared keys for various networks size

However, in a Distributed sensor network as in [4], if two nodes do not share a key they can still communicate using an indirect link (multi-hop). In an IoT network using RPL routing, multi hop alternative route is not possible. A node is only able to communicate with its preferred parent as per the routing table. In our experiment, if this node does not share a key with its preferred parent, then the link between those two nodes does not exist. Therefore the node will not be in the routing table and any sub leaves will also be discarded. fig. 3 show a simulation example of a 100 nodes network and how the routing table for a small subset of this network appear when simulated in the context of the Distributed Sensor Networks versus in the context of the Internet of Things. From this figure we can conclude that many nodes will be discarded if we use the key pre-distribution algorithm in its current form. This will result in an IoT network a lot smaller than the one we started with. The remaining nodes that were discarded, if the algorithm left as it is, will have to start the process of randomly selecting a new key ring and identifier ring. Nodes in the routing table will then check again whether all leaves in the routing table share a key.

4.3 Larger Key Rings

Having a small ring size for a considerably large network is a characteristic of the key pre-distribution algorithm in [4]. However and as shown in table 2, the rings size used for previous experiment did not achieve full connectivity of



(a) Network subset (b) Routing table for DSN (c) Routing table for IoT

Figure 3: Comparison of routing table for a snippets from a simulation of 100 nodes in the context of DSN Vs. IoT

Table 2: Simulation experiments over various rings size

Original values			Experiment											
			1		2		3		4		5		6	
N	RS	SK %	RS	SK %	RS	SK %	RS	SK %	RS	SK %	RS	SK %	RS	SK %
100	8	50.52	18	84.16	22	100								
250	13	50.43	30	98.18	36	100								
500	18	57.14	30	83.17	45	99.07	48	100						
750	22	49.47	30	71.95	45	92.87	60	99.40	63	100				
1,000	25	57.14	30	63.44	45	89.28	60	97.32	75	99.53	77	100		
2,500	41	48.19			45	59.37	60	92.46	75	97.11	100	99.64	104	100

the network. One alternative that we thought is essential to investigate is the size of the ring. Table 2 below show how we experimented with the ring size, modifying it until we reached 100% connectivity of the network.

Fig. 4 show a comparison of rings size when the key pre-distribution algorithm is used in distributed Sensors network and in RPL over IoT network for various network sizes. It is very clear that the size of the ring that achieves a full network connectivity in [4] does not apply to the Internet of Things network when using RPL. To achieve full connectivity of the network, a ring size of 77 key/identifier is needed for a pool size of 1000 in comparison of a ring size of 25 key/identifier for the same pool. This is a big difference that will have a large impact on the network performance. Fig. 5 show the rings size needed for various network sizes to achieve a guaranteed full connectivity between all nodes within the RPL routing table.

As we can see from table 2 above, 104 keys were needed in the key ring to achieve a 100% guaranteed connectivity in the RPL routing table in comparison with only 41 keys in a ring needed for DSN networks . We have used 64 bits key and 32 bits identifier. This will mean that key ring and identifier ring will take up around 1.38 kb of memory storage in each node. In this experiment, we have also used Zolertia node Z1 which features a powerful a 16-bit RISC CPU, 16MHz clock speed, 8KB RAM and a 92KB Flash memory. This means that at

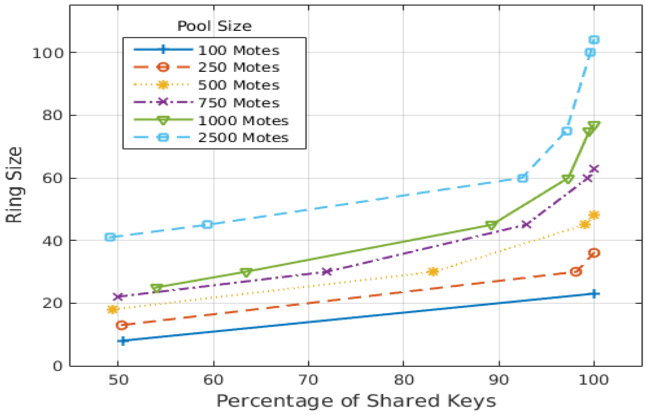


Figure 4: Various Rings size to achieve 100% of shared keys for different Pool size

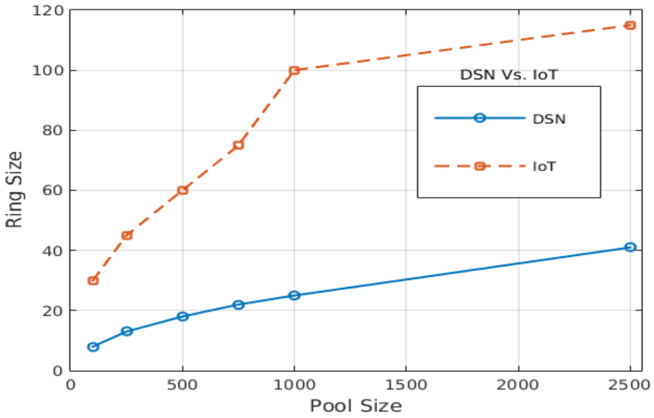


Figure 5: Rings size in DSN Vs. Rings size in IoT for various Pool size for 100% connectivity

least 90 kB of Flash memory is still free to use for operating system and other applications.

However, the original plan was to use as in [4] a pool of 100,000. A simple calculation can give us an estimation of 4,600 keys and identifiers in each ring in order to guarantee connectivity in the network using RPL protocol. Ring size of 4,600 keys and identifiers will take up around 54 kb of memory storage in each node. That is more than half of the memory present for the Zolertia node (Zolertia[18] has the largest amount of memory in Contiki. TMote sky node [19] is widely used and it has only 48 kb of memory which is not enough if using 4,600 keys and identifiers in each ring).

Computation overhead is another aspect that needs to be looked at. Comparing two identifiers rings will require a processing power that is very scarce. When running the same experiment using 4,600 and 104 keys in a ring, we note that during comparison of the key ring between two nodes, nodes processing power were around 87% used for 23 seconds. We can conclude that for a larger key ring size, nodes will not be able to cope with the computation power required and this will add a huge overhead on the network performance and the routing table establishment.

5 Conclusion and Future Work

In this paper, we investigated the performance of the key pre-distribution algorithm for distributed sensor networks on the IoT devices. We experimented with the variables and simulated small scale networks of 100 nodes to large scales network of 2500 nodes. Up until this point, we believe we have proved that the key pre-distribution algorithm achieve the 50% probability of the nodes to have a shared key, however it does not guarantee a full connectivity of the network when used in the context of the IoT. The use of RPL protocol in IoT gives a 0.45 probability of leaves in the RPL table with a shared key, which means that not all the network is able to communicate as the RPL only uses leaves that are in the routing table.

The next step in this research will be to explore alternatives solutions to secure leaves in the RPL routing table that do not share a key. In the coming few months, we will be developing a new Objective function metric.

The Objective Function uses several routing metrics to form the DODAG based on some algorithm or calculation formulas. Metrics are carried in DAG metric containers embedded in the DIO messages. The DAG metric containers at the moment are divided into two categories, node metrics and link metrics. In node metrics, nodes exchange information metrics about node state, node energy and hop count. in Link metric, nodes exchange link related information such as throughput, latency and link reliability.

We propose to add Shared Identifier Secure Link Objective Function (SISLOF) to RPL objective function metrics. SISLOF objective function will be used to quantify the shared key discovery (node metric) between two nodes that can form a direct link (neighbouring node) using a Boolean value, of 0 or 1, where 0 indicates that the two nodes do not share a common identifier and 1 indicates that the two nodes do share one or more common identifier. Further to this, the SISLOF will compute other link metrics in order to determine the suitability of the link if two links exist both with a shared key, in term of ETX and node rank.

By doing this, we ensure that any node that joins the routing table can communicate securely as only the nodes that fulfil the requirement of the SISLOF will be able to join the RPL DODAG.

References

[1] K. Zhao and L. Ge, A Survey on the Internet of Things Security, Computational Intelligence and Security (CIS), 2013 9th International Conference on, Leshan, 2013, pp. 663-667. doi: 10.1109/CIS.2013.145

[2] Lu Tan and Neng Wang, Future internet: The Internet of Things, 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), Chengdu, 2010, pp. V5-376-V5-380. doi: 10.1109/ICACTE.2010.5579543

[3] G. Gan, Z. Lu and J. Jiang, Internet of Things Security Analysis, Internet Technology and Applications (iTAP), 2011 International Conference on, Wuhan, 2011, pp. 1-4. doi: 10.1109/ITAP.2011.6006307

[4] Laurent Eschenauer and Virgil D. Gligor. 2002. A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM conference on Computer and communications security (CCS '02), Vijay Atluri (Ed.). ACM, New York, NY, USA, 41-47. DOI=http://dx.doi.org/10.1145/586110.586117

[5] Zach Shelby and Carsten Bormann, 6LoWPAN: The wireless embedded Internet - Part 1: Why 6LoWPAN?, EE Times (2011). http://www.eetimes.com/document.asp?doc_id=1278794

[6] IEEE Computer Society, 802.15.4 - Low Rate Wireless Personal Area Networks (LR-WPANs), IEEE standard for local and metropolitan area networks, IEEE, USA, 2011.

[7] Internet of Things, Strategic Research Roadmap; European Commission - Information Society and Media DG, European Commission, Brussels, Belgium, 2009.

[8] T. Winter, Ed.,P. Thubert, Ed., A. Brandt, J. Hui, R. Kelsey, , P. Levis, K. Pister, , R. Struik, , JP. Vasseur, and R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, IETF draft, 2012. https://tools.ietf.org/html/rfc6550

[9] T. Taso, R. Alexander, M. Dohler, V. Daza, A. Lozana and M. Richardson, Ed., A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs) RFC 7416, IETF trust (2015) https://tools.ietf.org/html/rfc7416

[10] D. W. Carman, P. S. Kruus and B. J. Matt, Constraints and Approaches for Distributed Sensor Network Security, dated September 1, 2000. NAI Labs Technical Report

[11] C. Neuman, T. Yu, S. Hartman, K. Raeburn. RFC 4129: The Kerberos Network Authentication Service, 2005.

[12] Noack, M., Optimization of Two-way Authentication Protocol in Internet of Things, Master Thesis, University of Zurich, Communication Systems Group, Department of Informatics, Zurich, Switzerland , 2014.

[13] P. Porambage, P. Kumar, A. Gurtov, M. Ylianttila and E. Harjula, Certificate based keying scheme for DTLS secured IoT draft-pporamba-dtls-certkey-00 , IETF, June 2013.

[14] Contiki Operating system http://contiki-os.org

[15] Frederik Ostrelind, A sensor Network Simulator for the Contiki OS, February 2006 http://soda.swedish-ict.se/2296/1/SICS-T–2006-05–SE.pdf.

[16] Claire, Swedberg, University Caters to Students Seeks Efficiencies Through Beacons, IoT Journal, Sep 2016. http://www.iotjournal.com/articles/view?14936

[17] IP Services, Birkbeck University of London. http://www.bbk.ac.uk/its/services
/kpis/wifi-usage, Last modified: Aug 23, 2016

[18] Zolertia Low power wireless module for IoT and WSN. http://zolertia.io/z1

[19] Sohpie Moore, Tmote Sky, August 2013. http://wirelesssensornetworks.weebly.com
/1/post/2013/08/tmote-sky.html

Secure routing in IoT networks with SISLOF

Ayman El Hajjar[1], George Roussos[1] and Maura Paterson[2]

[1]Department of Computer Science and Information Systems
[2]Department of Mathematics, Economics and Statistics
Birkbeck, University of London, London, UK
Email: [a.elhajjar, g.roussos, m.paterson] @bbk.ac.uk

***Abstract*—In this paper, we propose a modification of the RPL routing protocol by introducing the SISLOF Objective Function ensuring that only motes that share a suitable key can join the RPL routing table. This will ensure that all IoT network motes connect in a secure method. SISLOF uses the concept of key pre-distribution proposed by Eschenauer and Gligor in the context of the Internet of Things. First, we discuss related work that provide evidence that the key pre-distribution scheme in the context of the IoT with default RPL metrics fails to achieve the full network connectivity using the same ring size, however full time connectivity can be achieved but with a great cost in term of the large rings sizes. We introduce the SISLOF Objective Function and explain the modification it does to the RPL messages (DIO and DAO). We finally show the performance of the key pre-distribution in the context of the Internet of Things when SISLOF is used as the Objective Function of the RPL routing protocol.**

***Keywords*-Internet of Things; Security; RPL; Objective Function;**

I. Introduction

The Internet of Things (IoT) consists of things that are connected to the Internet, anytime, anywhere. It integrates sensors and devices into everyday objects that are connected to the Internet over fixed and wireless networks.

The Internet of Things will be made possible by using IP based network such as the IPv6 Low Wireless Personal Area Network (6LoWPAN). It is a simple low cost communication network that allows wireless connectivity in applications with limited power and relaxed throughput requirements [1]. The 6LoWPAN concept originated from the idea that “the Internet Protocol should be applied to low-power devices to participate in the Internet of Things [2].

The purpose of this paper is to propose an Objective Function (OF) called Shared Identifier Secure Link OF (SISLOF) for the Routing Protocol for Low Power and Lossy Networks that only adds to its routing table motes that share a key and thus can securley communicate.

The distribution of the keys to be used by SISLOF is based on Laurent Eschenauer and Virgil D. Gligor’s Algorithm [3] for Distributed Sensor Networks (DSN). We implement it in the context of 6LoWPAN Devices for the IoT. We provide an analysis of the performance of the SISLOF. We also compare its performance with the performance of the key pre distribution algorithm in the context of IoT with the default RPL routing metrics and in the context of DSN.

Section 2 provides an introduction to the Internet of Things, the 6LoWPAN network protocol, the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) and several solutions that attempts to secure the Internet of Things. Section 3 presents the key pre-distribution algorithm by Eschenauer and Gligor in [3] in the context of IoT when using the minimum ETX value (Default RPL metric) as in [4]. In section 4, we present the proposed SISLOF OF. In section 5 we provide an overview of the experiments setup and parameters used. In section 6 we provide an evaluation of the performance of SISLOF and how it compared with previous experiments. Finally, we present our main conclusions in Section 6.

II. Background Literature

Routing is a fundamental piece of the overall IPv6 architecture for the Internet of Things, and the Routing Protocol for Low Power and Lossy Networks, standardised as the the IPv6 routing protocol, is designed for large scale implementation of IPv6 in harsh environments that will translate the potential of Internet of Things into reality [5].

RPL organises its topology in a Directed Acyclic Graph (DAG). An RPL DAG must have at least one RPL root and a Destination Oriented DAG (DODAG) is constructed for each root. The root acts as a sink for the topology by storing all routes to all motes in the DODAG in the routing table [6]. For a DODAG to be constructed, the root will need first to broadcast a DODAG Information Object (DIO) message to all motes. The DIO message contains the DAG Metric Container option that is used to report metrics along the DODAG. Multiple metrics can be defined by an OF [6].

The OF is identified by an Objective Code Point (OCP) within the DIO Configuration option. An OF defines how motes translate one or more metrics and constraints, which are themselves defined in [7], into a value called Rank, which approximates the mote’s distance from a DODAG root in term of the number of hops it needs to reach it. An OF also defines how motes select parents. When a new DIO is received, the OF that corresponds to the Objective Code Point (OCP) in the DIO is triggered with the

978-1-5090-5873-0/17/$31.00 ©2017 IEEE

content of the DIO. For example, OF0 [8] is identified by OCP0 by the Internet Assigned Numbers Authority (IANA). The Minimum Rank with Hysteresis Objective Function (MRHOF) [9] is another OF defined by IANA and given the identifier OCP1.

Security specifically is a major issue as IEEE802.15.4 mandates link-layer security based on AES, but it omits any details about topics like bootstrapping, key management, and security at higher layers.

Security is Providing key management for confidentiality and group level authentication in a sensor network. The main challenge in public key algorithms when using in the context of Internet of Things, similarly to sensor networks, is the energy consumption of exchanging public key certificates [10] [11].

Key management protocols can be divided into three categories. Arbitrated keying protocols, Self Enforcing protocols and Pre-Deployed Keying protocols. Arbitrated keying protocols such as [12] and [13] are not suitable in the context of the Internet of Things because of the capabilities of sensor motes and leave the network vulnerable to man in the middle attacks. Self Enforcing protocols such as [14] to secure IoT was suggested to provide a lighter and robust security protocol using pairwise key establishment between motes however the communication overhead was considerably large. In the next section we show how the management scheme for Distributed Sensor Networks (DSN) proposed by Eschenauer and Gligor in [3] was used in the context of the IoT with the default RPL routing metric, the minimum ETX.

III. Previous Work

A. Key pre-Distribution Scheme

Offline Key pre-distribution algorithm for DSN proposed in [3] describes the method by which keys are distributed to motes in the network. This key pre-distribution mechanism ensures that for each direct link between any two motes in the network, the probability of those two sharing at least a key is 0.5. Using Stirling approximation, the authors of [3] concluded that the size of key rings KR does not need to be large in order for a network to guarantee full connectivity and only 50% of those motes need to have a shared key. An example in [3] showed that when a pool contained $100\,000$ keys, full network connectivity was achieved with only 75 keys in the rings.

This scheme was used in [4] in order to determine if it produces full connectivity in the context of the Internet of Things.

1) A large pool P of keys K are generated with their identifiers ID.
2) The Ring Size RS is equal for both keys rings $KR[RS]$ and identifiers rings $IR[RS]$.
3) Each identifier in $ID[RS]$ is of size b bits.
4) A mote send its identifier IR_s to another mote to establish if common identifiers exist with the receiver's identifier ring IR_r.
5) If a common identifier is found, the receiver sends back an acknowledgement with the identifier number i.e. "$ID_s[3]''$ to represent the third identifier in the identifier ring of the sender ID_s.
6) Once the sender receives the acknowledgement containing the common identifier found, a secure link is established using the key related to the identifier.

B. Performance of the Key Pre distribution Scheme in the context of the IoT with RPL using the Minimum ETX metric

Following the simulation of the Key Pre-distribution Scheme in [11] in the context of the Distributed Sensor Networks and using the minimum ETX value as the RPL metric to choose the preferred parent, the results of the simulation experiments showed that out of each pool used, only half of the leaves in the routing table shared a key. The other half was excluded from the RPL routing table. For example, the percentage of motes in the DODAG that has a shared key was 54.01% when the ring size RS was 25 keys in a pool P that contained a 1000 keys and a network of 1000 motes. Only when the ring size was increased to 77 keys that the full network connectivity was achieved and all motes in the network were included in the RPL routing table.

IV. SISLOF

The Shared Identifier Secure Link Objective Function (SISLOF) is our proposed OF to find secure links (those that share an identifier) between any mote and all of its candidate parents to form a secure RPL routing table while minimising the number of motes that are excluded because of insecure links.

SISLOF will attempt to find shared keys between motes by using the Key pre-distribution algorithm for Distributed Sensor Networks proposed in [3]. This will allow the formation of an RPL routing table that only contains secured links between motes.

A. Aims and Objectives

The aim of SISLOF is to create a secure RPL routing table with as many motes as possible. Specifically, its objectives are:

- Only motes that share a key can become a leaf in the DODAG tree.
- Nodes that do not share a key with their selected parent will discard this selection and try to form a leaf with one of the other motes that received its DIO (Neighbouring motes).
- If one mote shares a key with two or more motes, it will select as the preferred parent the mote that has a better ETX value in order to form the leaf between the two motes.

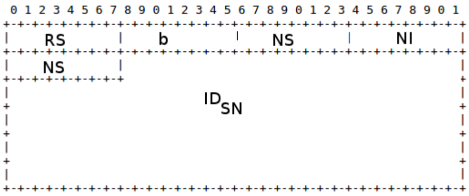


Figure 1. Addition to the DIO message: 1 byte for each of the variables, Ring Size (RS),identifier size (b), Number of identifiers in one message (NI) , Number of Sequence (NS) and Sequence Number (SN). ID_{SN} for the number of identifiers sent in the message.

B. SISLOF Metrics

SISLOF uses two types of metrics in its process to compute the preferred parent for a mote. First it uses our new mote metric object called Shared Identifiers State (SIS) to compare two arrays of identifiers in order to determine if one or more shared identifier exist.

If the mote that received the DIO determines that it shares one or more identifiers with two or more motes, that mote will need to choose which of the motes that sent the DIO will be selected as the preferred parent. SISLOF will thus need to decide between the motes it shares a key with. This will require SISLOF to use a link metric object as a second criterion in order to select the preferred parent. SISLOF will use the ETX Reliability object to select the preferred parent. The ETX value is calculated for each link from which a DIO message was received and with which it shares one or more identifiers. The mote that has the lowest ETX value will be selected as the preferred parent. The ETX is the number of transmissions the mote expects to make to a destination in order to successfully deliver the packet. This will also require changing the ‘A’ field of the header to 7 for each message (this field is given to indicate that the header will report a minimum or a maximum) [7].

Below is an explanation of the RPL messages modifications to incorporate the metrics required for the Key predistribution scheme by Eschenauer and Gligor in [3] as proposed by [4].

C. Message and Modifications

SISLOF will require the modification of the DIO and DAO RPL messages in order to encapsulate the various variables of SISLOF required to exchange identifier rings and look for a common one. Those variables will be either encapsulated in the DIO message sent to a mote or in the DAO message replying.

SISLOF variables shown in Fig. 1 and explained in Table I are composed mainly of identifiers and other values related to the segmentation of those identifiers. To incorporate the SISLOF variables shown in Table I in a DIO message, the 6LoWPAN message, the ICMPv6 control message and the DIO base object requires 89 bytes [15] which implies that there are 38 bytes in the data frame to be used to embed in frame variables related to SISLOF . In Fig. 1 RS and b are selected to fulfil requirements of the algorithm of [3]. NI provides the number of identifiers that can fit in the DIO payload. NI is calculated as the rounded integer of the available payload (33 bytes) divided by the identifier size b. NS is the total number of messages required to transmit the complete identifier ring. NS is calculated as the quotient of RS divided by NI. Finally SN identifies the order of the specific message in the complete sequence of messages required to disseminate the identifier ring. It is calculated as the sequence index corresponding to the current message.

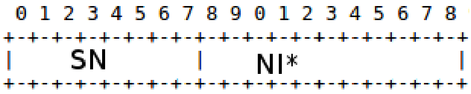
Table I
IDENTIFIER TRANSMISSION CONFIGURATION OPTIONS USED FOR TRANSFERING SISLOF MESSAGES IN DIO AND DAO.

Variable	Name of Field	Size in bytes
RS	Ring Size	1 byte
b	Identifier Size	1 byte
NI	Number of identifiers in one message	1 byte
NS	The Total Number of Sequences	1 byte
SN	The Sequence Number	1 byte

To encapsulate as many identifiers as possible in each DIO message, variables size in bytes are kept to the minimum by giving only 1 byte for each variable as shown in Table I. This means that each variable can have any value between 0 to 255 in decimal. Several factors were behind choosing these values. From experiments we did and using the same technique used in [3] with a 2500 mote network and the Ring Size RS that we used was 41 keys/identifiers for each ring. Using the same formula in [3] with the same network size and Pool size, the ring size for a network of 100 000 motes will be 250 keys. It can be represented in a 1 byte field. We have also used an Identifier Size of 1 byte. Using 1 byte for the Identifiers is more than enough, since the identifier is not used to encrypt the message and it is only used to identify if a common key exists between two motes. Using both RS and b will not yield a number of identifiers in one message larger than 256. In our example, using the same number of motes as [3] will yield one identifier NI per each message, that is 250 messages or the total number of sequences NS. The sequence number SN will of course be smaller than NS as it is a counter that will determine the sequence number of a specific message.

DAO messages takes 69 bytes as per [8]. This leaves us with 58 bytes in the data frame that we used to embed frames related to our OF used as below and shown in Fig. 2. SN is the sequence number received in the corresponding DIO. NI is a bitmap with bits set to 1 if the identifier with the corresponding position is available in the identifier ring of the mote that received the DIO message and 0 otherwise [1]. The DAO messages sent upward by each node that received the DIO is shown in Algorithm 1.

[1] NI size is variable and changes depending on the size of each identifier.



```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    SN         |    NI*              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. Addition to the DAO message. 1 byte for Sequence Number (SN) and NI, the bitmap representing shared identifiers bits.

D. Securing the link

A mote that is propagating the DODAG information, broadcasts the DIO message downwards. The DIO message will contain all information related to 6LoWPAN messages such as the IPv6 header, etc. On top of that, the DIO message will also contain its rank with the root. SISLOF addition to the DIO message, explained in Fig. 1 will contain the identifiers of the first DIO frame from the sequence of frames (NS).

One of the constraint variables that is required by the SISLOF is the shared identifier constraint. The calculation of this variable will produce a secure or insecure link. This variable will determine whether a mote is considered a secure candidate parent or not. This is the first constraint/criterion that SISLOF computes before moving to other variables to calculate the path between motes and the root and form the RPL routing table.

Each mote that receives a DIO message replies back with the DAO message the 6LoWPAN header. On top of that, the DAO message will also contain the SISLOF additions explained in Fig. 2.

Each node that receives a DIO message replies back with the DAO message that contains as of Fig. 4, all information related to 6LoWPAN message such as IPv6 header, etc. On top of that, the DAO message will also contain the SISLOF objective function additions explained in Fig. 2 The DAO messages sent upward by each node that received the DIO is shown in Algorithm 1.

The sequence diagram shown in Fig. 3 shows the various control messages and variables exchanged between two nodes in order to determine if a common identifier exists. After a common identifier is found, SISLOF will then compute the link metrics and the parent ETX in order to choose the preferred parent.

E. Link Metrics and parent ETX calculation.

If one or more secure mote that received the DIO identified that a shared identifier exist then the expected Transmission Count metric (ETX of the parent), similarly to the ETX calculation of RPL link metrics in [7], will become the second criteria on deciding the best parent. This metric will return the values of the DIO origin mote ETX ($parent\_metric$) and its received metric $instance\_etx$. From these two variables the link metric can be calculated to return the ETX of the link $link\_metric$ [15].

```
Input   :
  • DIO message ($DIO_{SN}$)
        $DIO_{SN}$=($n$, $b$, $IR_{SN}$, $NI$, $NS$, $SN$)
  • Identifier Ring of Receiver $IR_r$
    $IR_r = \big[ID_1,\ ID_2,\ ID_3,\ ID_4,\ \ldots\ ID_{(n-1)},\ ID_n\big]$
  • Ring Size (RS)
Output  :
  • Shared identifiers bits ($SIB_{SN}$)
    $SIB_{SN} = \big[b_1,\ b_2,\ b_3,\ b_4,\ \ldots\ b_{(NI-1)},\ b_{(NI)}\big]$
  • DAO message
        $DAO_{SN}$=($SN$), $SIB_{SN}$
  • Shared Identifier State ($SIS$)
    $SIS = \big[w_1,\ w_2,\ w_3,\ w_4,\ \ldots\ w_{(NI-1)},\ w_{(NI)}\big]$
$SIB_{SN} = [NI]$;
$x = 0$;
$y = 0$;
$z = 0$;
$w = 0$;
$SIS = [w]$;
for $w = 0$ to $RS - 1$ do
    for $y = 0$ to $NI - 1$ do
        for $z = 0$ to $RS - 1$ do
            if $IR_{SN}[y] = IR_r[z]$ then
                Append 0 To $SIB_{SN}$;
                $SIS[w] = 0$ ;
            else
                Append 1 To $SIB_{SN}$;
                $SIS[w] = 1$ ;
        end
    end
end
AddtoDictionary $DAO_{SN}$ ($SIB_{SN}$ "Shared
 Identifiers bits", ($SN$) "Sequence Number" );
Send $DAO_{SN}$ upward to DIO Sender ;
```

Algorithm 1: DAO Messages Algorithm

DAO messages each a reply to a DIO message from the sequence it receives, contains a bitmap stream of bits representing either a value of 1 for a shared identifier and a value of 0 for a not shared identifier in SIB_{SN} for all identifiers in the ring of the receive mote.

V. Experiment Setup and Parameters

Similarly to the experiments carried on in [4] and [11], the experiments were simulated using the Cooja application in the Contiki Operating System.

A C program was coded to implement the key pre-distribution algorithm of [3]. This resulted in the generation

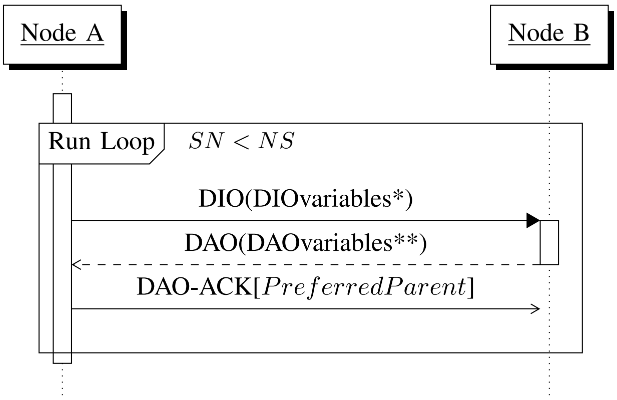


Figure 3. SISLOF Sequence Diagram
diovariables*: RI , IS, Num.Of.Seq , Num.Of.Iden, ID_{SN}[], Seq.Num
daovariables:** Seq.Num,NI_{SN}[], ETX

of Keys Pool, IDs pool, Key rings and ID rings [2].

The parameters used in the SISLOF experiment are the same as in [4] and [11]. The overall area of the simulation was kept to 250x250 meters, a typical size of a medium size university [3]. The transmitting range for each mote is set to 50 meters (this is the common transmitting range for 6LoWPAN low power devices). We also used the key length $klength$ of 64 bits and the ID length $ilength$ of 32 bits.v The Pool size P for both keys and identifiers is the first parameter. The pools size we run simulations for are: 100, 250, 500, 750, 1 000 and 2 500 motes. The second parameter is the network size N.

In this paper we are looking at the maximum number of motes as it is an important factor to determine the number of keys shared between motes in comparison to it. The third parameter is the ring size RS For each pool size (P), keys and identifiers and to ensure the accuracy of experiment simulations, each experiment was run 5 times with the largest and smallest results discarded and the average of the remaining three runs used. In our experiments, if this node does not share a key with its preferred parent, then the link between those two nodes does not exist. Therefore the node will not be in the routing table and any sub leaves will also be discarded. In addition to this, when simulating smaller number and given that the simulation area is not changed, the number Percentage of Shared Keys (SK %) for 10 or 25 motes in the network is low as motes are unable to communicate with each other since the network motes are sparse.

VI. Results

The proposed Objective Function SISLOF was simulated using the the parameters explained in the previous section. This presented us with three different sets of experiments, the first in [4] where the pre key distribution scheme was simulated in the context of Wireless Sensor Networks. The second in [11] where the scheme was simulated in the context of the IoT using the default RPL routing metric, the Minimum Expected Transmission Count ETX. The third is the simulation where the scheme is simulated in the context of the IoT using SISLOF for RPL. The number of keys in the ring size RS for each of the three set of experiments is shown in Table II below with the percentage of Shared Keys (SK %) between motes that formed leaves in the routing table.

Table II
Comparison table showing precentage of Shared Keys (SK%) when original Ring Size (RS) and Network Size (N) are used, when Minimum ETX metric is used and when SISLOF metrics are used.

Original values			Experiment			
			Minimum ETX metric [4]		SISLOF	
N	RS	SK %	RS	SK %	RS	SK %
100	8	50.52	23	100	12	100
250	13	50.43	36	100	20	100
500	18	57.14	48	100	28	100
750	22	49.47	63	100	38	100
1000	25	57.14	77	100	40	100
2500	41	48.19	104	100	60	100

From Table II, we can notice that the ring sizes in DSN was quite low in comparison with the ring sizes for IoT when the Minimum ETX metric was used. However it is also clear that the ring sizes when SISLOF is used, is around 55% less then when RPL was using with the ETX metric. From Fig. 4, we can observe the performance of the key pre-distribution using the three experiment sets results presented in the table. The key-pre-distribution in the DSN networks presented the lowest ring sizes and the IoT using the Minimum ETX metric for RPL showed the highest ring sizes.

Wireless Sensor Networks required the smallest ring sizes to achieve full connectivity simply because in DSN a mote that do not share a key with one of its neighbours can send data to that specific neighbour indirectly through another mote and thus the full network connectivity is achieved even if not all motes share keys.

The ring size needed to achieve full connectivity when RPL was used with its default minimum ETX metric was the largest because only motes that share a key can participate in the RPL routing table. Nodes that did not share key could not communicate. By increasing the size of the ring, we ensured in [11] that all motes can join the RPL routing table and thus communicate.

From [4] and [11], we identified that 104 keys and identifiers in the rings was needed to achieve a 100% guaranteed connectivity in the network comparison with only 60 keys when SISLOF was used. Using the parameters we explained in section 5, we can conclude that the key ring and the identifier ring in each mote for a network of 2 500 motes will

[2] Different random generators were used for keys, IDS and pools [4]

[3] Birkbeck, University of London [11]

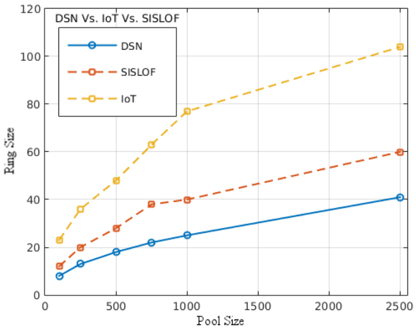


Figure 4. Comparison of the rings sizes used in Key Pre Distribution Scheme (DSN, IoT and RPL with SISLOF).

take up around $0.72kb$. This is an actual saving of nearly 50% in term of capacity in comparison with the required storage of 1.38 kb for a 104 key ring and a 104 identifier ring. In this experiment, we have used Zolertia mote Z1 which features a 92KB Flash memory. This means that more than 90 kB of Flash memory is still free to use for other applications. Using the calculation as of [3], we can expect the ring sizes for $100\,000$ to be in the region of 2400. This will require around 28.8 kB of Flash memory.

VII. Conclusion

In this document we proposed Shared the Identifier Secure Link Objective Function (SISLOF), an Objective Function that identifies motes that share a secure links in the network and uses secure links as the first criterion for calculating the RPL routing table.

We have investigated the performance of SISLOF and its impact on the security of an Internet of Things network. The results of the rings sizes in the SISLOF experiments is clearly a lot smaller then the rings sizes in the IoT experiments. We have provided evidence that by using SISLOF we can secure all communications between motes in the Internet of Things as only motes that share a key can be joined in the routing table and thus all communications between motes are secure.

The experiments simulated indicate that by using SISLOF, the ring size in term of number of keys and identifiers in comparison with the size of ring size wwhen using RPL with minimum ETX metric was nearly half. This resulted in a reduction of storage compairson to nearly half as well. Those savings will also have a direct impact on the power consumption. Less keys and identifiers in the ring will also result in less messages being exchanged between motes and thus using less battery power.

The proposed SISLOF provides evidence that it is able to secure the IoT in an efficient way for small area such a medium size university, however more research is required in order to determine its suitability in term of the overhead it generates in the network when RPL messages are propagating to all motes to form the routing table and the storage space it will consume once networks become larger. One possible solution that is worth exploring is to have multiple DODAGs with secure routes between roots.

References

[1] Z. Shelby and C. Bormann, *6LoWPAN The Wireless Embedded Internet*, 1st ed. Wiley, 2007.

[2] I. C. Society, "802.15.4 low rate wireless personal area networks (lr-wpans)," 2011.

[3] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM CCS*. NY, USA: ACM, 2002, pp. 41–47.

[4] A. E. Hajjar, G.Roussos, and M. Paterson, "Securing the internet of things devices using pre-distributed keys," in *IC2EW*, April 2016, pp. 198–200.

[5] "Jp vasseur, milestone in connecting the internet of things – rpl routing standard completed," cisco, 2012.

[6] T. Winter, P. Thubert, and et.al, "Rpl: Ipv6 routing protocol for low-power and lossy networks," RFC 6550, March 2012.

[7] J. Vasseur, M. Kim, and et.al, "Routing metrics used for path calculation in low-power and lossy networks," RFC 6551, March 2012.

[8] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (rpl)," RFC 6552, March 2012.

[9] O. Gnawali and P. Levis, "The minimum rank with hysteresis objective function," RFC 6719, September 2012.

[10] D. W. Carman, Kruus, and et.al, "Constraints and approaches for distributed sensor network security (final)," *DARPA Project report,(Cryptographic Technologies Group, Trusted Information System, NAI Labs)*, vol. 1, no. 1, 2000.

[11] A. E. Hajjar, G.Roussos, and M. Paterson, "On the performance of key pre-distribution for rpl-based iot networks," in *3rd EAI International Conference on Safety and Security in Internet of Things*, October 2016.

[12] R. Mukundan, K. Morneault, and N. Mangalpally, "Digital private network signaling system (dpnss)," Internet Requests for Comments, RFC 4129, September 2005.

[13] M. Noack, "Optimization of two-way authentication protocol in internet of things."

[14] D. A. Ha, Nguyen, and et.al, "Efficient authentication of resource-constrained iot devices based on ecqv implicit certificates and datagram transport layer security protocol," in *The 7th proceedings of SoICT*. NY, USA: ACM, 2016, pp. 173–179.

[15] J. Hui and J. Vasseur, "The routing protocol for low-power and lossy networks (rpl) option for carrying rpl information in data-plane datagrams," RFC 6553, March 2012.