

Optimized Approach for Isolation of Version Number Attack in IoT Systems

by

Parthkumar Patel
202011062

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY



May,2022

Declaration

I hereby declare that

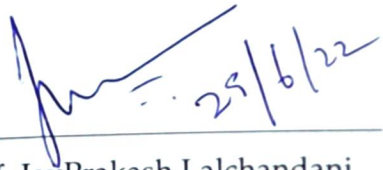
- i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,
- ii) due acknowledgment has been made in the text to all the reference material used.



Parthkumar Patel

Certificate

This is to certify that the thesis work entitled optimized approach for isolation of version number attack in IoT systems has been carried out by Parthkumar Patel for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.



Prof. JayPrakash Lalchandani
Thesis Supervisor

Acknowledgments

I would like to take this opportunity to express a sense of gratitude and indebtedness to Prof. JayPrakash Lalchandani for his constant support and encouragement at the various stages of this thesis work. I express my deep sense of gratitude to respected Prof. Saurabh Tiwari, M.Tech conveyor, for his kind cooperation. I would also like to thank M.Tech ICT program coordinator for his guidance.

I must express my gratitude to my parents and family members for being with me at every moment and providing continuous morale boost and affection during my thesis work. I must add that it has been a great experience studying at DAIICT, and I will always cherish the haloed memory of my alma mater.

Lastly, I thank the almighty for always being there for me.

Contents

Abstract	v
List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 IoT Security Attacks	1
1.2 Motivation	2
1.3 Objectives	3
1.4 Organisation of the Thesis	3
1.5 Chapter Summary	4
2 IoT Attacks and its Consequences	5
2.1 Types of IoT Attacks	5
2.1.1 Physical Attacks	5
2.1.2 Network Attacks	7
2.1.3 Software Attacks	7
2.1.4 Encryption Attacks	8
2.2 Literature Review	8
2.2.1 Summary of Literature Review	12
2.2.2 Existing Techniques	15
2.3 Chapter Summary	17
3 RPL Protocol and its Security Mechanism	18
3.1 DODAG in RPL	18
3.2 Version Number Attack Threat	20
3.3 RPL Security	21
3.4 Global Repair Mechanism	22
3.5 Chapter Summary	22

4	Proposed Work	23
4.1	Problem Formulation	23
4.2	Proposed Method	23
4.2.1	Implementation Strategy	24
4.2.2	Proposed Technique	26
4.2.3	Simulation Setup	26
4.2.4	Parameter for Performance Analysis	27
4.3	Chapter Summary	28
5	Results and Discussions	29
5.1	Simulation Results	30
5.2	Comparison with Existing Methods	38
5.3	Chapter Summary	39
6	Conclusion	40
	References	41
	Appendix A Installation and Setting up for Network Simulator 2	44

Abstract

Internet of Things (IoT) involves deploying low power and lossy networks, allowing communications among pervasive devices such as embedded sensors. A different network may have different requirements, and hence the Routing protocol for Low-Power and Lossy Networks (RPL) protocol supports different routing topologies called Destination Oriented Directed Acyclic Graph (DODAG) to optimize the routing. The version number system of DODAG is used to verify that the topology is optimal. An attacker may present in topology and has the ability to exploit the network topology mechanism by altering the version number and hence may reduce its lifetime. In this study, different types of attacks on IoT systems are discussed. A review of the RPL protocol and DODAG functionality is conducted, as well as two strategies for isolating the version number attack in IoT. A technique for detection and isolation of multiple version number attacks is discussed. In this research work, an optimized technique is proposed and is implemented in network simulator 2. The outcomes of the simulation were examined in terms of message overhead, throughput, delay, and power consumption.

Keywords: Internet of Things(IoT), version number attack, IoT attacks, network simulator.

List of Tables

- 4.1 Simulation parameters 27
- 5.1 Network performance with different attackers 34
- 5.2 Average of network performance parameters 35
- 5.3 Average of network performance parameters by increasing nodes . 35

List of Figures

2.1	Elimination technique	16
2.2	Shield technique:	17
3.1	DODAG system build	19
3.2	Reordering of DODAG	20
4.1	Schematic diagram for overall working of proposed model	25
5.1	Simulation topology	29
5.2	Throughput analysis	30
5.3	Delay analysis	31
5.4	Overhead analysis	32
5.5	Power consumption analysis	33
5.6	Packet loss analysis	34
5.7	Topology with 50 nodes	36
5.8	Topology with 70 nodes	36
5.9	Topology with 90 nodes	37
5.10	Topology with 110 nodes	37
A.1	Installing essential libraries	44
A.2	Appending code to sources.list	45
A.3	Update linux packages	45
A.4	Setting path using .barshrc	47

CHAPTER 1

Introduction

The internet of things(IoT) is a technology in which multiple sensors, intelligent nodes, and devices are bound to one another in order to communicate without the use of human work. Object function works independently depending on the link between them.

IoT nodes execute tasks such as analyzing acquired data for decision making, providing lightweight data, and gathering and extracting data by accessing and authenticating cloud-based resources. Through the internet of things, people, services, sensors, and objects are all inextricably linked.

IoT devices are used in various applications, from smart cities to intelligent transportation systems. Because of the significant commercial opportunities afforded by IoT scenarios, the number of smart devices and intelligent services provided through IoT networks has been outgrowing. [1].Since IoT devices are dependent on cloud infrastructure, cloud-based IoT networks have been built so that data may be transported among applications.

An IoT device contains vast amounts of data, much of which is unique to its individual users, including online browsing/purchase records, credit card details and personal health information.An improperly secured device leaves this data vulnerable to theft. What's more, vulnerable devices can be used as gateways to other areas of the network they are deployed on, allowing for more sensitive data to be extracted.

1.1 IoT Security Attacks

Various security vulnerabilities and threats exist in IoT systems, which are covered in the following chapter.

Physical, network, software, and encryption attacks are the four major categories into which the various attacks are classified.

The internet of things is implemented using a variety of existing network technologies.

As a result, correct classification is required to contain various forms of attacks, which can also be useful in creating and implementing ways to secure IoT devices. The sheer volume of Internet of Things devices makes their security a high priority and is crucial for the future wellbeing of the internet ecosystem. For device users, this means abiding by basic security best practices, such as changing default security passwords and blocking unnecessary remote access (e.g., when not required for a device's functionality). Vendors and device manufacturers, on the other hand, should take a broader approach and invest heavily in securing IoT management tools. Steps that should be taken include:

- Proactively notifying users about devices running outdated software/OS versions.
- Enforcing smart password management (e.g., mandatory default password changes).
- Disabling remote access to a device, unless it's necessary for core functions.
- Introducing a strict access control policy for APIs.

1.2 Motivation

The IoT network is much vulnerable to various type of security attacks. IoT involves deploying low power and lossy networks, allowing communications among pervasive devices such as embedded sensors. A different network may have different requirements, and hence RPL protocol supports different routing topologies called DODAG to optimize the routing. Each DODAG is assigned a version number. The purpose of the version number is to ensure that there are loop free paths to the root node, the routing table entries of nodes in the DODAG are not obsolete and there is no inconsistency in the DODAG. The root node in a DODAG increments the version number in case of any inconsistency. This calls for a global repair process and the DAG is reconstructed. A malicious node may advertise a false version number in its control message to force a global repair. A technique for detection and isolation of multiple version number attacks should be proposed.

1.3 Objectives

- To test the performance of DODAG protocol under the impact of version number attack in IoT devices.
- To Implement version number attack mitigation technique in IoT.
- Design an optimized technique for the identifying of version number attack in IoT based devices on No. of packets forwarded by the node.
- Implement a designed scheme to compare it with an already existing scheme for the detection of version number attack in IoT devices.

Attack was triggered on network topology with variable number of nodes and variable number of attacker nodes in NS-2 and its impact is recorded and are tabulated later in chapter 5. Optimized approach for mitigation of such attacks is described in chapter 4 and the same is implemented and compared with existing techniques like shield in chapter 5.

1.4 Organisation of the Thesis

First chapter of the thesis introduces IoT systems and how IoT devices are connected to form a network. It briefly describes the various possible attacks possible among the IoT networks.

The second chapter discusses various types of attacks possible in IoT systems based on the nature of the attack. It includes extensive literature review and comparison among various research publications. It also discusses two existing technique for isolating version number attack namely, shield and elimination.

The third chapter gives the whole idea of the RPL protocol proposed by IETF, how the nodes in this protocol arrange themselves to form a unique topology and how version number attack is carried out because of the presence of a loophole in the global repair mechanism.

Chapter four discusses the proposed work. It also describes the problem formulation. It shows implementation strategy for overall working model and discusses the schematic diagram for the same. The chapter also includes simulation setup and discusses what parameters are to be used for performance analysis.

Chapter five includes results from the simulation and its effects on changing the topology, the number of attacker nodes, and a different number of nodes in the

topology. Chapter six includes the conclusion of the thesis. In the end, Appendix-A comprises setting up and installing ns-2 for simulation purpose.

1.5 Chapter Summary

Chapter 1 introduces the IoT technology. A brief introduction to the IoT security attack is also presented, along with possible types of attacks. The chapter also discusses the motivation behind the thesis work. The chapter also includes a guide to objectives and organisation of the thesis.

CHAPTER 2

IoT Attacks and its Consequences

The explosion of IoT technologies incited users and organizations to swiftly adopt IoT devices to enhance process control and boost productivity. The rise of connected devices has transformed the way users' data is processed and stored. Since IoT devices are smart devices and often interact with other devices over the internet, the personal information they collect makes them vulnerable to various security risks. According to a survey, 84 percent of organizations have deployed IoT devices on their corporate networks, and more than 50 percent don't maintain the necessary security measures beyond default passwords. Cyber criminals often rely on IoT connections to compromise network systems and steal personal information. Unpatched vulnerabilities and manufacturing defects in connected devices become a gateway for threat actors to penetrate corporate networks.

2.1 Types of IoT Attacks

It is estimated that with the rise in number of things connected to IoT systems to swarming billions of devices, the potential vulnerabilities will also increase. Hence, the increase in vulnerabilities due to loophole in IoT technologies may give rise to security incidents in IoT systems. Some of the most common security issues in IoT are highlighted in succeeding subsections.

2.1.1 Physical Attacks

The hardware devices of IoT systems are a main target in these types of attacks. The existence of an attacker close or even within the system is decisive for guaranteeing that attacks take place. This category also includes attacks that threaten the lifespan or functionality of hardware.

The following are a few of these attacks:

1. Node Jamming in WSNs: The physical attack of radio frequency interference and the node jamming attack have certain similarities. The attacker has the ability to disrupt the rf signals of nodes in WSNs. As a result, the signals may be jammed, blocking communication between the nodes. When the attacker blocks the key sensor nodes, the IoT service can be successfully blocked.
2. RF Interference on RFIDs: Noise signals are created and broadcast through radio frequency signal to carry out a DoS attack on an RFID tag. When noise signals interfere with RFID signals, the connection will be disrupted.
3. Node Tampering: A sensor device can be harmed by an attacker by replacing the whole node or a portion of its physical devices. The node can be electronically interrogated, allowing the malicious node to access or change crucial information.
4. Malicious Code Injection: The attacker injects a real node into the malicious node to compromise its security. As a result, this attack can gain access to the IoT system. The attacker has complete control over the node or the entire system.
5. Sleep Deprivation Attack: The majority of sensor nodes used in IoT systems have replaceable batteries. By configuring and following sleep routines, the battery performance is extended. The nodes are awoken throughout this attack, which increases power consumption and causes the nodes to die in a short period of time.
6. Malicious Node Injection: A new malicious node can be deployed by the adversary among two or more IoT system nodes. As a result, the data flow between the nodes and their actions are managed. The MITM (Man in the Middle) Attack is another name for this strategy.
7. Social Engineering: The attacker manipulates IoT users to obtain confidential information or execute specific actions that aid in attaining the attacker's goals. This attack is classified as a physical attack since it necessitates physical interaction between the attacker and IoT users on the network.
8. Physical Damage: A malicious node might cause harm to IoT devices for personal gain. This form of attack secures the region or building that contains an IoT system. The adversary damages the IoT system directly, causing service availability to be affected in the existence of the attack, this way it is different from node tampering attack.

2.1.2 Network Attacks

Such attacks are present in the IoT system network. It is not necessary for the malicious node to be close to the network in order to carry out an attack.

1. **RFID Cloning:** An attacker copies the data from a victim's RFID tag to some other RFID tag in order to clone it. Despite the fact that the two RFID tags have the same data, the original RFID ID is not reproduced. As a result, it is easy to distinguish between original and hacked data.
2. **Traffic Analysis Attacks:** The fact that wireless qualities makes it possible for an attacker to obtain secret data that is passed over RFID devices. In almost all attacks, the attacker extracts some network information at the start using sniffer apps.
3. **RFID Spoofing:** An attacker spoofs RFID signals in order to read and record data transmissions from RFID tags. The attacker then sends the personal information together with the actual tag ID. Given the presence of the original tag ID, the system believes this data to be genuine. As a result, gaining full access to the network becomes simple for the attacker.
4. **RFID Unauthorised Access:** Due to the lack of suitable authentication procedures in most RFID systems, this becomes easy for anybody to access tags. As a result, data stored on RFID devices can be read, updated, and even deleted [5].
5. **Sinkhole Attack:** The attacker is attracting all traffic from WSN nodes and creates a metaphorical sinkhole. As a result, all packets routed to a certain destination are dropped, resulting in data confidentiality being compromised and network services being denied.
6. **Sybil Attack:** Single malicious node impersonates the identities of a huge number of nodes, and this attack is known as the sybil attack. In this form of attack, the nearby nodes accept fake information. A sybil node can be selected within the routing path, and it can also vote several times in the WSN voting system.

2.1.3 Software Attacks

Web application and related software vulnerabilities in IoT devices can damage systems on whole. Due to the presence of software attacks, many harmful pro-

grams are used in attempts to alter or steal the information and potentially endanger to IoT devices.

1. Phishing Attacks: The attacker uses phishing website emails to mimic the user's authentic credentials in order to get access to confidential information.
2. Malicious Scripts: When IoT network's internet connection is up and running, data can be lost, or the entire system can be shut down if active-x scripts are supplied to the gateway that is in charge independently by users for execution.
3. Denial of Service: A denial-of-service attack is a cyber-attack in which the attacker attempts to render a computer or network resource unavailable to its intended users by temporarily or forever disrupting the services of a network host. The attacker has complete access to the application layer due to the blocking of legitimate users from the application layer.

2.1.4 Encryption Attacks

Such attacks target the encryption techniques used within the IoT system. Breaking the system encryption is the goal of encryption attacks.

1. Cryptanalysis Attacks: In these attacks, a ciphertext or plaintext is expected to be present. By identifying the encryption key utilised in the systems, the encryption mechanism of the system is broken.
2. Side-channel Attacks: The attacker obtains the encryption key used for data encryption and decryption by employing certain techniques on the encrypted devices found in IoT systems.

2.2 Literature Review

A source location protection mechanism was proposed by Guangjie Han et al. [9]. His technique is solely based on dynamic routing, which resolves privacy concerns connected to the source location. The goal of this strategy is to expand the length of the data transmission line. This approach chooses a starting node at random from the network's perimeter. Before reaching the sink, all packed data will pass via greedy nodes and follow the remaining directed path. Several tests

were carried out in order to confirm the theoretical findings. Findings from the experiments show that the suggested technique can safeguard the network lifetime while preserving source location privacy and fighting against various privacy disclosure attacks such as eavesdropping security breaches, direction-focused assaults, and so on. As a result, the researcher determined that by regulating the energy consumption rate, the proposed technique successfully improves the network lifetime.

A new was proposed by Ugur Bekcibasi et al. [4] to improve localization accuracy. The use of a variable-length reference anchor technique can improve accuracy. Simulations can be used to calculate the effectiveness of the proposed method, and the results can then be analyzed and compared. In order to assess this enhanced performance, three-node conventional localization models were used. Both models have been tested in three different places, and statistical results have been discovered. As a result, the researcher has come to certain conclusions that suggest that the suggested technique improves RSSI localization performance significantly. The researcher also determined that by employing this strategy, the energy usage rate will be reduced, resulting in a longer network lifetime.

A greedy method that uses compact attack graphs to secure IoT systems is suggested by Beytullah Yigita et al. [20]. In the beginning, all viable attack paths that reach the network's predetermined resources are retrieved. Furthermore, the original condition with the lowest practical cost is eliminated. The cost function computation contributes to the attack routes and elimination cost. This process repeats in an iterative fashion until the condition of total cost exceeding the allocated budget is met. According to the studies from the experiment, the algorithm scales slowly with increasing the network size. This proposed approach may be applied well to large-scale networks with a huge deployment of IoT nodes.

Y. Liu et al. [21] described a software-defined networking-based solution to defending against network assaults (SDN). In order to safeguard data and the internet of things devices, a model was created utilizing middlebox-guard (MG). This model secures traffic flow at several points by following various procedures and policies. This model also includes an integer linear programming (ILP) algorithm for switching volume limits and existing traffic interruption. This formula is capable of balancing the load and dealing with a variety of challenges. This helps in balancing the traffic flow in the network. As a result, data transfer was managed safely, and the security speed was improved.

D.Yin et al. [6] described a method to detect DDoS attacks using the software-based internet of things (SB-IoT). The SB-IoT is composed of various parts that work together to operate IoT devices. That comprises gateways, controllers, and devices built on the internet of things platform (IoT). A threshold value is also used to protect against attacks from susceptible and heterogeneous devices. This number indicates the real-time occurrence of a DDoS attack as well as the origin of the attacker. Finally, the implementation technique is used to determine DDoS susceptibility in a shorter amount of time and revamps the obvious vulnerability.

R. H. Jhaveri [12] introduced a framework for combating mobile ad-hoc network security assaults. By removing the network's malfunctioned packets, this solution tackles the problem of security. As a result, the traffic flow is monitored and analyzed by a trust route (TRS) using pattern discovery (PD). In addition, three other metrics were used to assess packet dribble attacks. The network simulator two was used to determine the precise value of the parameter by controlling the data packets and control packet drop ratio, which gives the threshold level by using detection rate. The created method replaces the faulty value with a positive one.

Shailendra Rathore, et al. [18] proposed a novel ESFCM technique along with the fog-based attack detection approach, which is based on fog computing architecture. Stretching cloud computing and recognition of distributed attacks allows for threat detection at the edge of the network, as well as the discovery of spread attacks. The labelled data problem is controlled by utilizing a fuzzy c-means technique that is semi-supervised. For providing a better generalization performance at higher detection rate, an algorithm based on the extreme learning machine (ELM) is provided. In terms of accuracy and attack spotting time, it is clear that the suggested approach outperforms previous alternatives.

Yongfeng Qian, et al. [16] presented an analysis of various security issues being faced within the three different layers of IoT. Further, the blockchain based IoT security system is designed in this paper. The management of assets for devices present within full life cycles depending upon the blockchain is the major design issue of this security approach. The management needs of IoT devices are studied here and then the issues are solved by applying blockchain platform. For ensuring security and reliability, a device identification-based key algorithm is applied in case when the IoT devices and blockchain database interact with each other. Monitoring unusual network traffic which is on the basis of machine learning as well as the verification of identity are the two open issues discussed here.

Security fusion as a Service is a revolutionary concept introduced by Chien-Ting Kuo et al. [13]. (SFaaS). Models of attack are detected by integrating two detection approaches on a software switch topology measuring design scenario. The mechanisms, as well as the software measurement design service, are evaluated, analyzed, and simulated within this paper. The validity of the SFaaS approach is proved due to the high-performance results achieved for detecting and reducing the damage within these systems.

Rizwan Hamid Randhawa, et al. [17] proposed a cross-layer technique through which the CCM occurring on OSCoAP is removed. In this technique, the mac-layer security design within the IoT devices is utilized to exploit CCM. Mostly, the 802.15.4 radio chips are available within the devices present within IoT. The CCM is included within the radio chips to add certain security features for the mac-layer encryption by including the IEEE standard. The CCM operations for OSCoAP are implemented by considering the benefits of the features provided on the present devices. The life of the battery is enhanced along with the speed and memory efficiency as per the simulation results achieved by implementing the proposed approach.

Jaegeun Moon, et al. [15] proposed a novel public key cryptography technique which uses the lattice-based cryptography which is implemented by proposing Ring-LWE mechanism. It is unavoidable to carry out optimization in order to apply the strategy to IoT devices. The vulnerability towards side-channel attacks is higher in case of Ring-LWE approach. It can be observed from the outcomes of the experiments that vulnerability of existing additive modulus operators is high along with the possibility of leaking private keys.

Peiyuan Sun, et al. [18] designed a new method through which the activities of the attacker can be modeled using a machine learning approach. The observation that temporally closer attacks are launched by the attackers that exist within a similar botnet, is the base on which the activities of the attacker are modeled. A special class of point process is then utilized to model the attack temporal patterns. Further, the mutually exciting characteristics are used to recognize the latent influences amongst the attackers. The inferred weighted influence matrix is used along with graph-based clustering techniques to cluster the activities of the attacker. The activity pattern as well as the structure of botnets is recognized effectively as per the achieved results.

Shamsul Huda, et al. [10] proposed a detection model on the basis of SCADA network traffic which helped in designing a secure architecture for the ICS network. Two ensemble based detection algorithms are also developed in the proposed design. Instead of using conventional techniques, the network traffic and payload features are applied within the novel design approach to create detection models. The real SCADA network data is used in this paper for the verification of proposed ICS architecture. It is seen through the simulation results that in comparison to existing approaches, the proposed detection system performs better.

Mohamed Tahar Hammi, et al. [8] proposed a new mechanism through which the devices can be identified and authenticated in a robust manner. The bubbles of trust are generated here which are basically the original decentralized system. Also, the integrity and availability of data is protected through this approach. The security benefits that are achieved through the block-chains are a major factor considered in the proposed approach. The scenario in which the objects can identify and trust each other are created as secure virtual zones also known as the bubbles. It is seen through the achieved results that along with high efficiency and low cost, this approach provides IoT security.

2.2.1 Summary of Literature Review

Author's Name	Technique	Advantages/ Features	Disadvantages/ Improvements
Guangjie Han et al.	The author proposes a source location protection mechanism that is solely based on dynamic routing and tackles source location privacy concerns.[9]	The suggested system may secure the network lifetime while preserving source location privacy and combating various privacy disclosure attacks such as eavesdropping attacks, direction oriented assaults, and so on.	The security level provided by this approach is not very efficient even though it provides strong SLP.

Ugur Bekcibasi et al.	A new technique for improving localization accuracy has been proposed. [4]	The rate of energy consumption will decrease, extending the network lifetime.	Extra hardware and software are required in this proposed technique.
[20] Beytullah Yigita, et al.	A greedy technique is given in which compact attack graphs are used to secure IoT systems.	This proposed approach may be applied well to large-scale networks with a large number of IoT nodes.	The security metrics are not calculated in terms of various parameters such as integrity, confidentiality and availability of IoT assets.
[21] Y. Liu, et al.	To safeguard data and the internet of things, a model was created utilising middlebox-guard (MG).	The data flow was managed safely, and the security's speed was improved.	Not all attacks could be identified through this implemented approach.
[6] D.Yin, et al.	The approach for detecting a DDoS assault using software-based internet of things (SB-IoT).	The implemented approach determines DDoS vulnerability in a shorter amount of time and improves the obvious vulnerability.	The proactive handling of DDoS attacks within SD-IoT is not explained here.
[12] R. H. Jhaveri	The traffic flow was monitored and analysed using a trusted routing system (TRS) with pattern discovery (PD).	The developed approach updates the malfunctioned value with the positive one.	The requirement of hardware increased its cost.

[18] Shailendra Rathore, et al.	A novel ESFCM technique was proposed along with the fog-based attack detection approach which is based on architecture based on computing based on fog.	In terms of accuracy and attack detection time, the suggested approach outperforms previous alternatives.	The random assignment of input bias and weights causes the performance outcomes to be poorer.
[16] Yongfeng Qian, et al.	The block-chain based IoT security system is designed in this publication.	Network traffic which is abnormal is monitoring which is based on machine learning as well as the verification of identity are the two open issues discussed here.	The time consumed by the proposed approach was very high.
[13] Chien-Ting Kuo, et al.	A novel concept security fusion as a service is the name of the service (SFaaS) is proposed.	The validity of SFaaS approach is proved due to the high performance results achieved for detecting and reducing the damage within these systems.	The selection and detection of the correct applicant has a low hit ratio.
[17] Rizwan Hamid Randhawa, et al.	A cross-layer technique is proposed through which the CCM occurring on OSCoAP is removed.	The life of the battery is enhanced along with the speed and memory efficiency as per the simulation results achieved by implementing the proposed approach.	There is no improvement done for network and transport layers in IoT.

[15] Jaegeun Moon, et al.	Novel public key cryptography technique is proposed which uses the lattice-based cryptography which is implemented by proposing Ring-LWE mechanism.	The vulnerability of existing additive modulus operators is high along with the risk of a private key being leaked.	The proposed technique is not limited to 8-bit controllers; it can be used for controllers with more bits.
[19] Peiyuan Sun, et al.	A new method is proposed through which the activities of the attacker can be modelled using a machine learning approach.	The activity pattern as well as the structure of botnets is recognized effectively as per the achieved results.	The attackers launch the attack in an asynchronous way, which may influence the accuracy of the attack modelling.
[10] Shamsul Huda, et al.	Detection model on the basis of SCADA network traffic is proposed which helps in designing For the ICS network, a secure architecture is required.	As may be observed from the simulation results, in comparison to existing approaches, the proposed detection system performs better.	In our proposed technique, DBN training does not take place in real time.
[8] Mohamed Tahar Hammi, et al.	A new mechanism is proposed through which the devices can be identified and authenticated in a robust manner.	With high efficiency and low cost, this approach provides IoT security.	For the compromised devices, no solution is provided here.

2.2.2 Existing Techniques

1. Elimination Technique:

In the case of the elimination technique, Every node keeps track of all incoming messages and DIO messages. As discussed earlier, the DIO message has all the information about the topology and the changes to be made in the topology. It contains the version number which is responsible for start-

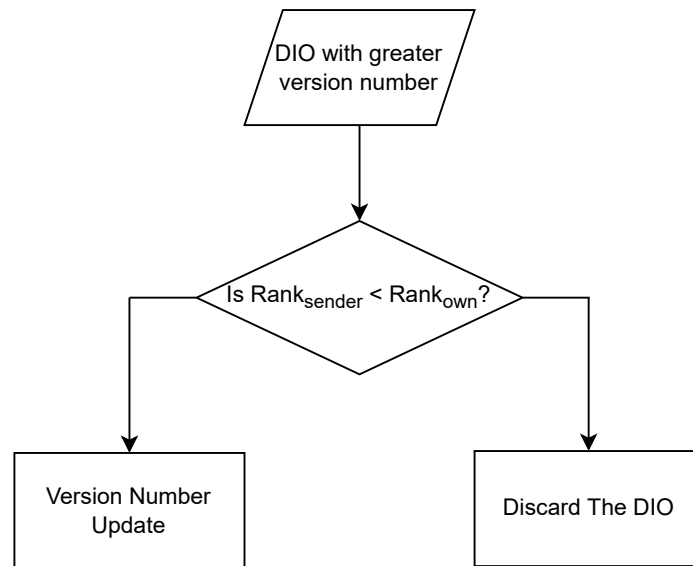


Figure 2.1: Elimination technique

ing the global repair mechanism. So every node checks the DIO message and compares the version number. If the version number of the sender is greater than the version number of its own, then this DIO message will be discarded. In other case, Version number will be allowed to update.

2. **Shield Technique:** Shield technique work similar to the elimination technique with additional security filter which is applied on the next step based on shieldlist. DIO messages have all the information about the topology and the changes to be made in the topology. It contains the version number which is responsible for starting the global repair mechanism. Initially node will check if the sender of DIO messages there in the shieldlist or not, if not then discard the upcoming DIO message. If yes then update that version number in the shield list only. Then next step is based on election where it is checked that if half of the nodes in shieldlist has the same version number then only final updation of version number will take place or else the DIO message will be discarded at the second step.

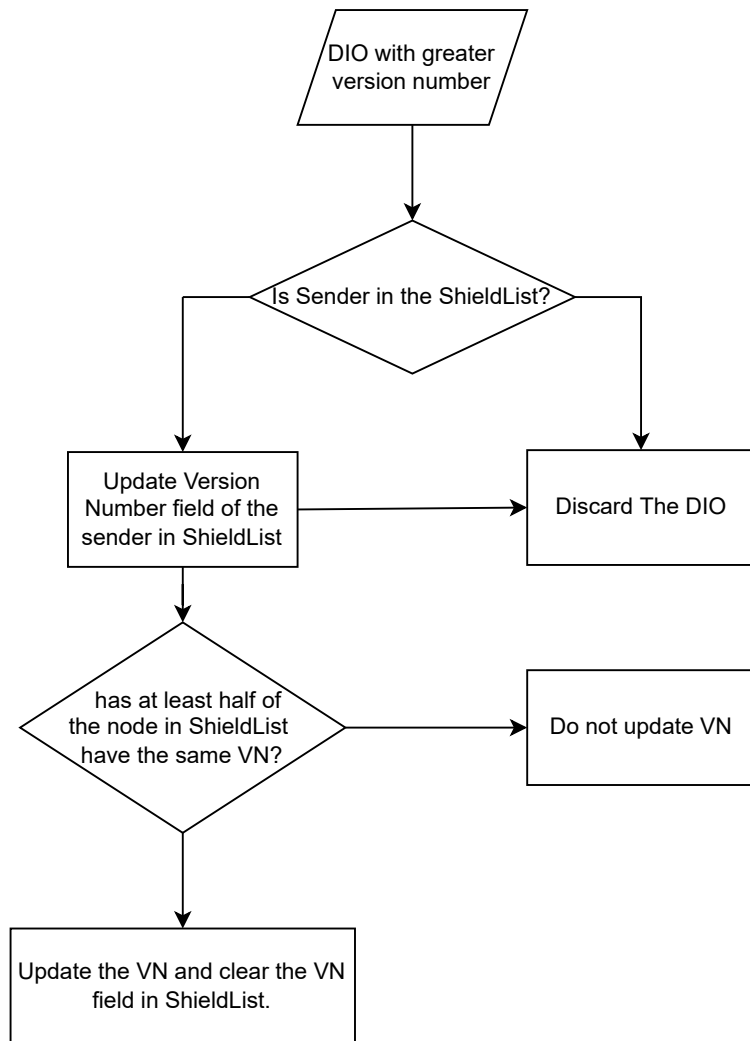


Figure 2.2: Shield technique:

2.3 Chapter Summary

Chapter 2 introduces the various possible attacks in IoT. A brief introduction to physical, network, software and encryption attacks is provided. The chapter also discusses the literature survey along with its summary in tabular form. It compares the major technique, advantages and disadvantages of that technique. The chapter also includes two existing techniques called elimination and shield.

CHAPTER 3

RPL Protocol and its Security Mechanism

Lossy networks and low power (LLNs) are a type of network infrastructure in which network devices are resource constrained (limited computing, storing, communications, and power sources) and the environment is lossy. When the embedded system's storage capacity is restricted, data of such low-power devices can readily be lost in the network.[1]LLNs are used in a wide range of contexts, from simple temperature measurements to high-volume multimedia services that require reliable communication support. It is frequently used to monitor the environment for the smart house, building automation, urban sensors, etc. It is also widely used in asset tracking, industrial control, and healthcare. RPL is vulnerable to attacks like cloneid, version number, DODAG inconsistency, and rank manipulation, to name a few.[1]RPL global repair mechanism is found to be very costly, which in turn drains limited energy available in such a resource-constrained environment. RPL is a distance vector routing protocol, The term distance vector refers to the fact that the protocol manipulates vectors (arrays) of distances to other nodes in the network. It is inter domain routing protocol and requires router to inform its neighbours when there is change in topology. RPL is also source routing protocol. It allows the sender of packet to partially or completely specify the packet route through the network which enables a node to discover all the possible routes to a host.

3.1 DODAG in RPL

A DAG (directed acyclic graph) with no outgoing edges that are rooted at a single destination called DAG root (DODAG root). [7] The DODAG rank represents a node's location in reference to other nodes as well as the DODAG root. DIO (DODAG Information Object) message is multicasted downwards. This message can be broadcast by a node in a DODAG, letting other nodes know about it. It advertises the message to another node If they want to join the topology.

DIS(DODAG information solicitation), This message is sent by a node that wants to join a DODAG.A request submitted by a child to a parent or root is known as a DODAG Advertisement Object.[14] It allows the child to join a DODAG.

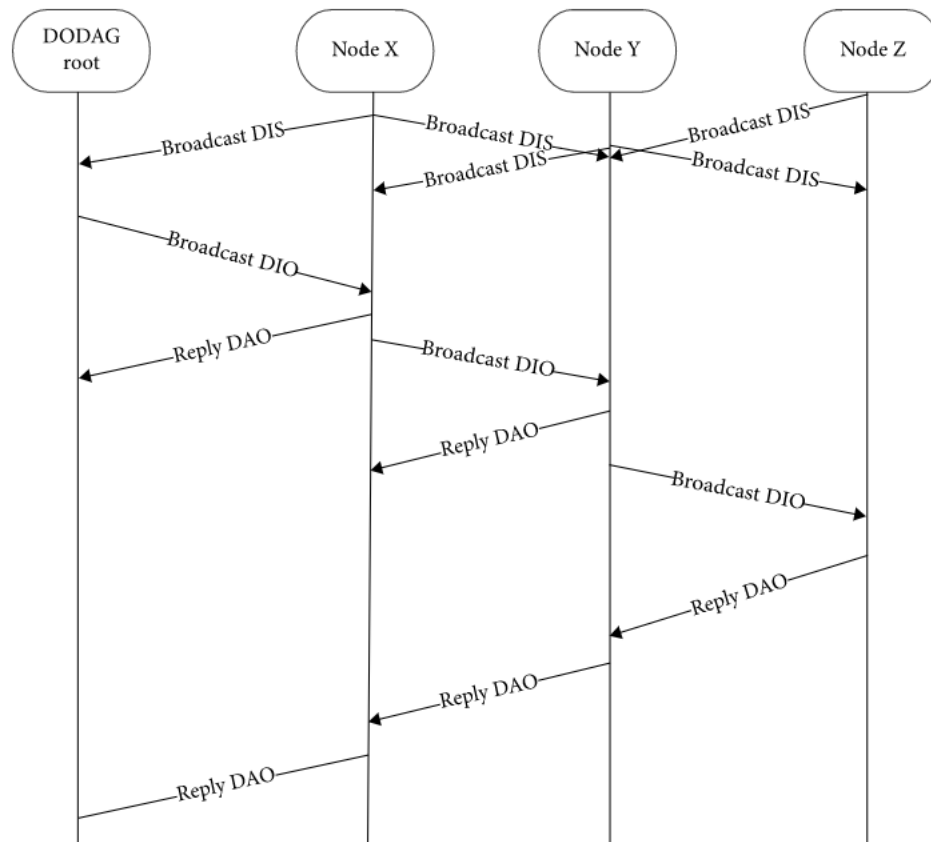


Figure 3.1: DODAG system build

If the node does not join any DODAG system, nor does it receive any DIO message, it periodically loops a DIS message to its direct-connected neighbor, requests DODAG information from the surrounding direct-connected node, computes it, and so on, until the node joins a DODAG system.

The DODAG root node broadcasts the DIO message carrying the DODAG information. The node X receives the DIO message broadcasted by the root node, joins the DODAG system, and after processing the DAO message with the node X prefix information passes it to the DODAG root node. DODAG node X sends DIO message containing DODAG information to the node directly connected with node X. After receiving the message, node Y directly connected with node X joins the DODAG system and replies the DAO message to node X, node X becomes the parent of node Y. Node Y receives the DODAG information request information DIS from node Z, but node Y will not reply to any information until node Y joins a

DODAG system. After node Y joins a DODAG system, node Y sends a DIO message to node Z and invites node Z to join the DODAG system. After receiving the message, Z, which is directly connected with Y node, joins the DODAG system and replies the DAO message to Y node, and Y node becomes the parent of Z. After receiving the message from node Z, node Y adds its own routing information and fuses it into the received message and then sends the DAO message to the preferred parent node X directly with node Y. By this analogy, the DODAG root node receives and calculates the DAO messages from each node in the DODAG system [12] and obtains the prefix information of each node in the DODAG system step by step.

3.2 Version Number Attack Threat

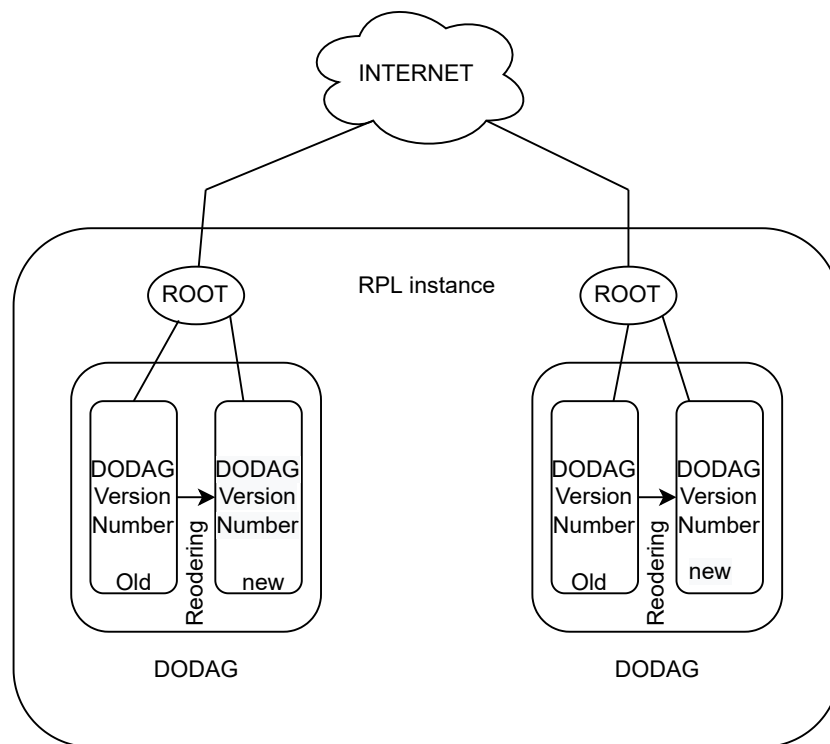


Figure 3.2: Reordering of DODAG

Since RPL is resource-constrained and the environment is lossy. The RPL organises resource-constrained nodes within an LLN using a destination oriented directed acyclic graph (DODAG) structure.[2] The root of the DODAG is a border router that is constantly working as a gateway. (Fig. 3.1) Meaning of version number DODAG, control messages are utilized to construct parent-child connections.

Two procedures, global and local repair, ensure the maintenance of an optimal DODAG in RPL. That means the global repair process rebuilds the entire DODAG from the ground up and affects the entire LLN. By incrementing a specific number known as the version number, the root starts the global repair. The DODAG is rebuilt (as represented in Fig 3.1) at the end of the operation, and the update is communicated by the RPL control message DIO. Just the root does have the authority to change the VN and, as a result, start RPL's global repair mechanism. When a malicious node updates the VN field in its DIO messages, causing an unauthorised global repair, the other nodes participate in, exchange control messages and re-establishing parent and child ties. This form of assault is known as a Version Number Attack (VNA).[2] VNA represents a significant threat to RPL-based IoT infrastructure and systems because it causes an inordinate amount of control packets to be swapped by IoT nodes, and the loss of the majority of application packets, resulting in long delays and increased energy consumption, all of which can lessen the lifetime of the network, reliability, and availability.[2]

3.3 RPL Security

Secure routing protocols for data packets in IoT networks is a challenging issue due to the characteristics that are inherited from other networks. Data packet routing in IoT-constrained devices suffer from potential security threats, and this has a considerable impact since it is related to the users. Several RPL attacks occur through the activities of malicious nodes during the data packet routing among devices. The RPL security has been extensively reviewed in [8]. Various kinds of RPL attacks have been analyzed, yet most of the studies have not concentrated on the mechanisms of secure RPL. In recent years, efforts have been made to design secure routing protocols for IoT-constrained devices. However, they all rely on conventional cryptographic functions, which drastically drain the resources of devices and impact the performance of the constrained devices likely to be used in IoT applications. The RPL characteristics, such as a lack of infrastructure, unreliable links, resource constraints, limited physical security, and dynamic topology, make them vulnerable and hard to protect against attacks. The data traffic density in certain nodes can lead to depleting batteries and resources faster than other nodes, which deactivates the routing data to the root node. Due to the billions of interconnected devices on the network, securing and protecting them from different forms of attacks creates a critical challenge. When these devices are vulnerable to attacks, such as rank attacks and version number attacks, users will feel that

their data are insecure.

3.4 Global Repair Mechanism

In RPL, the global repair mechanism is used as a maintenance mechanism to keep the DODAG healthy and optimal.[2] It attempts to rebuild the entire DODAG from the ground up and has an impact on the entire LLN. The root increments the VN, which initiates the global repair. Rebuilding of DODAG is done at end, and the update is transmitted by DIO messages. During this procedure, nodes re-assess their ranks, recreate parent-child connections, and exchange several control messages. According the RPL specification, only the root has the ability to refresh the VN and start the global repair procedure.[2] Furthermore, the specification makes no recommendations as to when or under what conditions such an operation should be performed, leaving the discretion of the implementers. On the other side, an attacker node can deliberately change the VN in RPL VNA and perform an unlawful global repair, causing the rest of the nodes to reconstruct the DODAG.

3.5 Chapter Summary

Chapter 3 explains the overall concept of IEFT's RPL protocol. It shows how the nodes of network forms a topology called DODAG and the message transfer takes place when a node joins the topology. It examines the way version number attack is triggered in the network and its impact on network performance. It shows how the reordering takes place in RPL instance when there is an update in version number. Chapter explains RPL security and its protection against version number attack. In the end of chapter global repair mechanism is explained in brief.

CHAPTER 4

Proposed Work

The two step technique for mitigating the version number attack was proposed in IoT. In the first technique, the virtual number attackers are mitigated based on the direction of leaf nodes. The version number attackers which are remained in the network will be mitigated with the second technique. In the second technique, the node will not update its virtual number until it has majority vote.[11] The proposed technique does not detect the multiple version number attack from the network. The mitigation scheme can be tested over the sensor devices which have low battery power and resources. An efficient technique for isolating the attack is proposed which used less resources in terms of power consumption etc. It is four step technique and are described in detail in following subsections.

4.1 Problem Formulation

The IoT network is much vulnerable to various type of security attacks. The DODAG is a hierarchical topology which is used in RPL for small devices. The version number attack is triggered by the attacker and it increments the version number which leads to establishment of path with loop. A malicious node may advertise a false version number in its control message to force a global repair. There is a need of technique for detection and isolation of version number attack from network. The IoT data is very critical to transmit this type of information to destination efficient path is required from one end to another. The technique of path optimization needs to apply which can improve path reliability in the network.

4.2 Proposed Method

An optimized approach for isolating the version number attack is proposed. The method works on the basis of number of packets forwarded by the node in the

topology. Network with finite number of nodes are deployed. Version number attack is then triggered into network. Based on number of packets transmitted to actual number of packets reached the destination, malicious node from the network is identified and eventually isolated from the network. Network performance is analyzed at the end. following subsection implementation strategy describes all the steps in detail.

4.2.1 Implementation Strategy

The following subsection describe the implementation strategy. It is mainly divided into four parts as described below:

1. Deployment of the Network Topology: The network will be deployed with a finite number of sensor devices and with the base station. The sensor devices are responsible to sense various types of conditions like temperature, pressure, etc. The sensor devices are heterogeneous means sensors have different battery and processing power. The DODAG protocol will arrange the network into a tree-like structure.
2. Trigger of Version Number Attack:: The malicious nodes will be formed in the network which is responsible to trigger the VNA. The DODAG protocol's version number will be updated by the malicious nodes. (The DODAG protocol will select the path which has a high version number in the network, this results in the construction of loop-based pathways.
3. Trust Calculation: In this study, a trust-based approach is developed for the isolating the version number attacks. The strategy built on trust will work in the three phases which are pre-processing, trust calculation, and trust updation. The sensor devices which have the least trust will be marked as malicious from the network.
4. Analyze Network Performance: In the last phase, the network performance will be analyzed in terms of certain parameters like throughput, packet loss, and energy consumption, to check the effectiveness of the proposed technique.

Fig 4.1 is a schematic representation for overall working of proposed model in which NS2 will provide the interfacing medium to the user. Once the four steps as

described above gets completed, the next step includes the analysis and comparison of the results to other proposed technique in terms of network performance parameters.

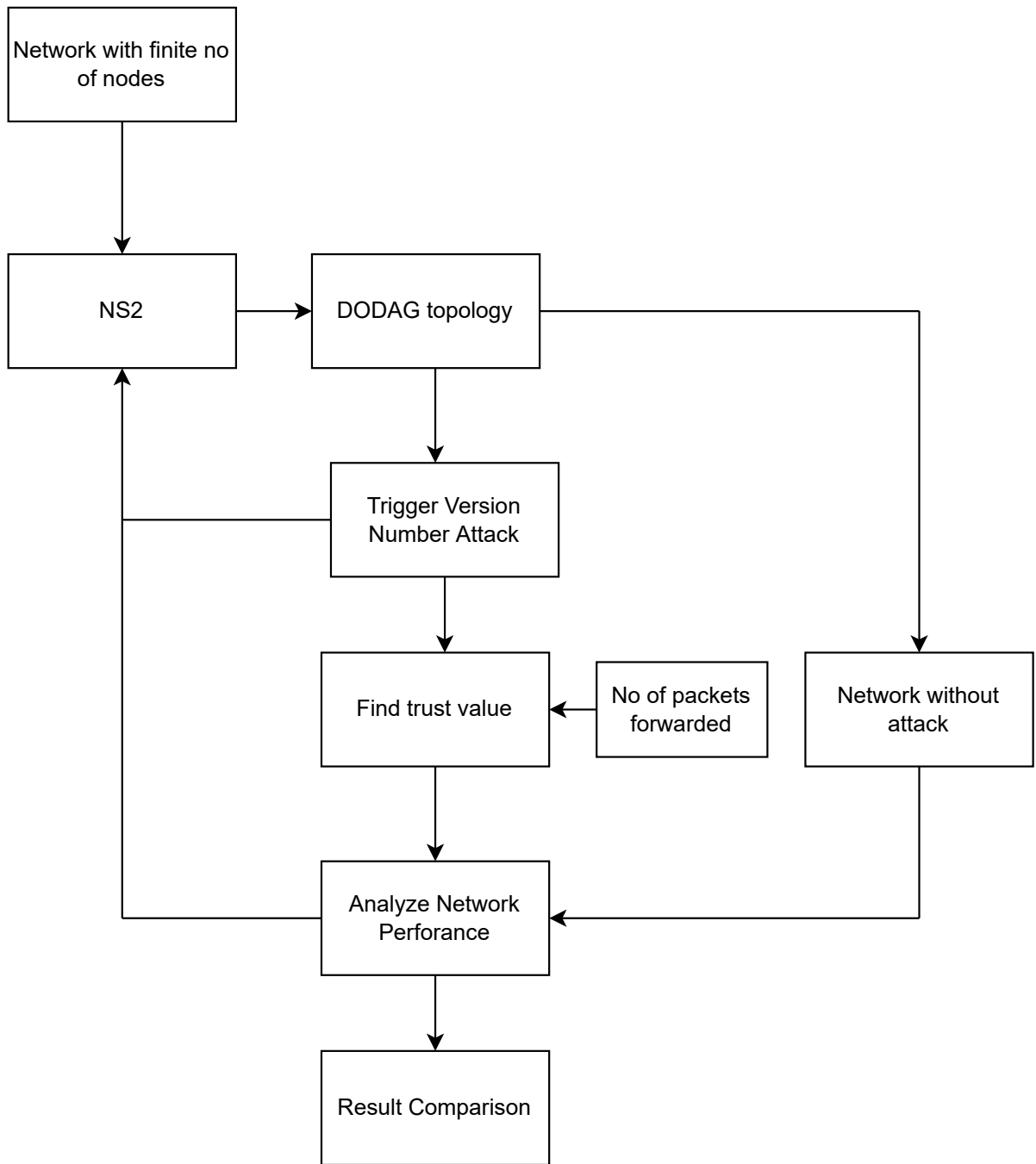


Figure 4.1: Schematic diagram for overall working of proposed model

4.2.2 Proposed Technique

Algorithm 1: Proposed Algorithm to detect attacker node.

Input: Deployed network with nodes representing sensor

Output: Identification of attacker node from network.

- 1 Network topology deployment using finite nodes representing sensors.
 - 2 Split the topology in a fixed size cluster.
 - 3 Select head of each cluster depending on the energy usage its distance.
 - 4 Find-out value of factor trust.
 - (I) Determine the quantity of packet sent from sensor node.
 - (II) Determine the total quantity of packet forwarded.
 - (III) PDR represents the total quantity of packets sent through network by the source node.
 - 5 If (Threshold value > PDR)
 - (I) Declare node as attacker.
 - 6 Setup route from the origin to the destination.
 - 7 Send a packet over a new path.
 - 8 End
-

4.2.3 Simulation Setup

Developing an IoT network in the actual world is extremely difficult in the field of IoT research, even an one testbed takes a lot of time and costs a lot of money. Fortunately, using simulation tools, we can immediately receive an analysis, track the progress, and assess the model's security and safety. Researchers could quickly setup, build, and operate the nodes using scripts. Various simulation tools, on the other hand, have different properties. Simulation is to be performed on unix based operating system with a minimum disk space of 400 MB and 8 gb of ram. In this study, NS2 will be utilized as a simulator to verify the technique's reliability. In most cases, research involves creating new modules in C++ and compiling them into the system. ns-2 is used at a high level. Our needs are met by the stock ns-2 distribution. The "all-in-one" release, which still needs compilation but includes a simple install script, is probably the simplest way to get ns-2 up and

running. (The ns-alone-2.35 version seems to require g++ or gcc versions which is not older than 4.7 for the purpose of compilation.) The native environment of ns-2 is linux. For windows users, the simplest way is to set up an unix virtual machine and then after install ns-2 inside of it. It is also possible to compile ns-2 on a cygwin machine.

Simulation Tool	NS2
Simulation Runtime	> 60 minutes
Total no of Nodes	upto 110
Size of Node(x)	30<x<55px
Normal Nodes	upto 109
Minimum Distance between Nodes	5 m
VNA Nodes	2
Area	800*800

Table 4.1: Simulation parameters

4.2.4 Parameter for Performance Analysis

- **Throughput:-** It is a parameter that is employed in the performance analysis. It indicates the quantity of packets successfully arrived at the destination in a certain amount of time.
- **Packet Loss:** Packet loss happens when one or more data packets travelling over a computer network fail to complete their intended destination. Data transmission issues can cause packet loss, or a combination of circumstances might cause packet loss.
- **Energy Consumption:** The entire energy consumed by the network for transmission, reception, and data aggregation is referred to as energy consumption.
- **Control Packet Overhead:** It's an indirect processing time that includes memory, bandwidth, and other resources used to complete a task.
- **Delay:** The latency of a network reflects how long it would take for a bit of information to go from one node or destination to another throughout the network.

4.3 Chapter Summary

In chapter 4, proposed work is presented. Based on the problem formulation, objectives are discussed. Thereafter, the implementation strategy is framed and a schematic diagram for the same is created. Furthermore, simulation setup and the parameters that are used for performance analysis are determined and discussed.

CHAPTER 5

Results and Discussions

In order to fix challenges and achieve efficiently, we must test and develop research approaches before exposing them to the real world. Fortunately, using simulation tools, we can immediately receive an analysis, track the progress, and assess the model's security and safety. Developing an IoT network in the real world is exceedingly difficult in the field of IoT research, even one testbed takes a lot of time and costs a lot of money.

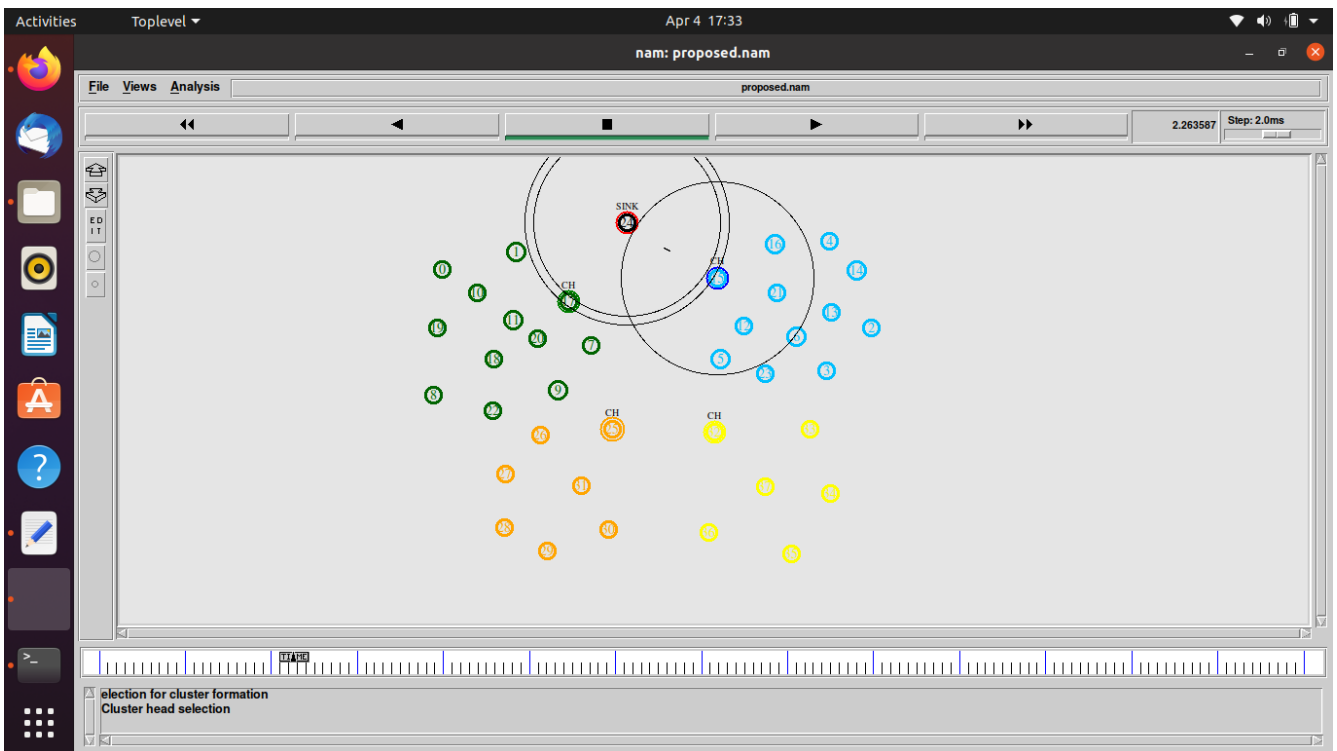


Figure 5.1: Simulation topology

5.1 Simulation Results

Due to the presence of malicious nodes in the network, the attack model result leads to higher packet loss and much more collisions in the network, which in turn lowers the number of packets arriving their destination and, as a result, a reduced throughput. As the time increases, the proposed technique behaves better in terms of throughput.

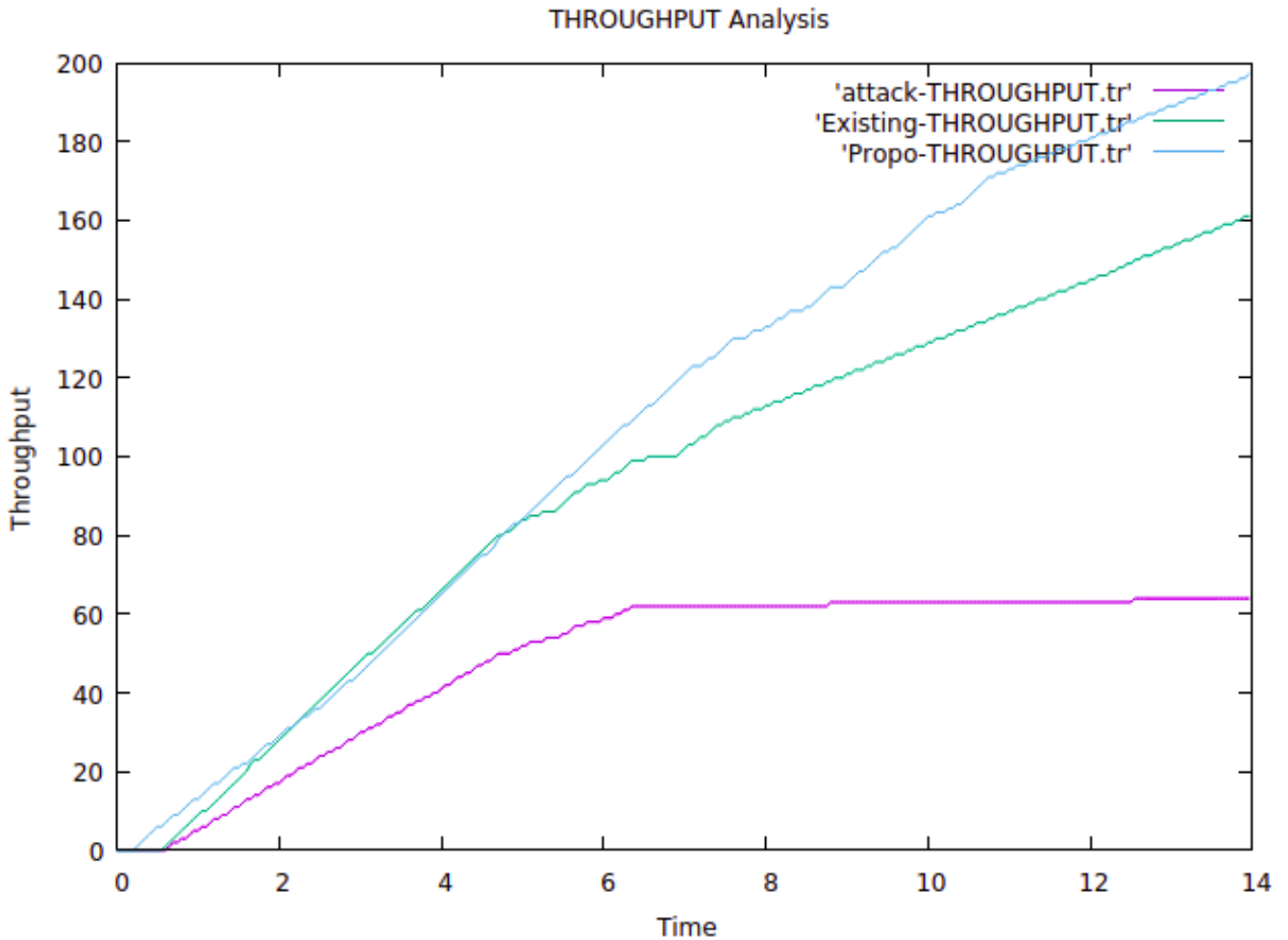


Figure 5.2: Throughput analysis

As we know that delay is the time it requires for a data packet to reach from one node to another node. As we see in the above fig 5.3 graph that violet line shows the delay in terms of attack scenario. When compared to the existing and proposed techniques, the highest amount of packets are dropped, and the delay is larger in

case of attacker mode. In the case of an attack scenario, attack nodes in the network cause packets to be dropped or placed on hold for longer periods of time, resulting in needless retransmissions in the network and greater delay, whereas in the proposed technique, a more reliable channel is chosen to send the packets.

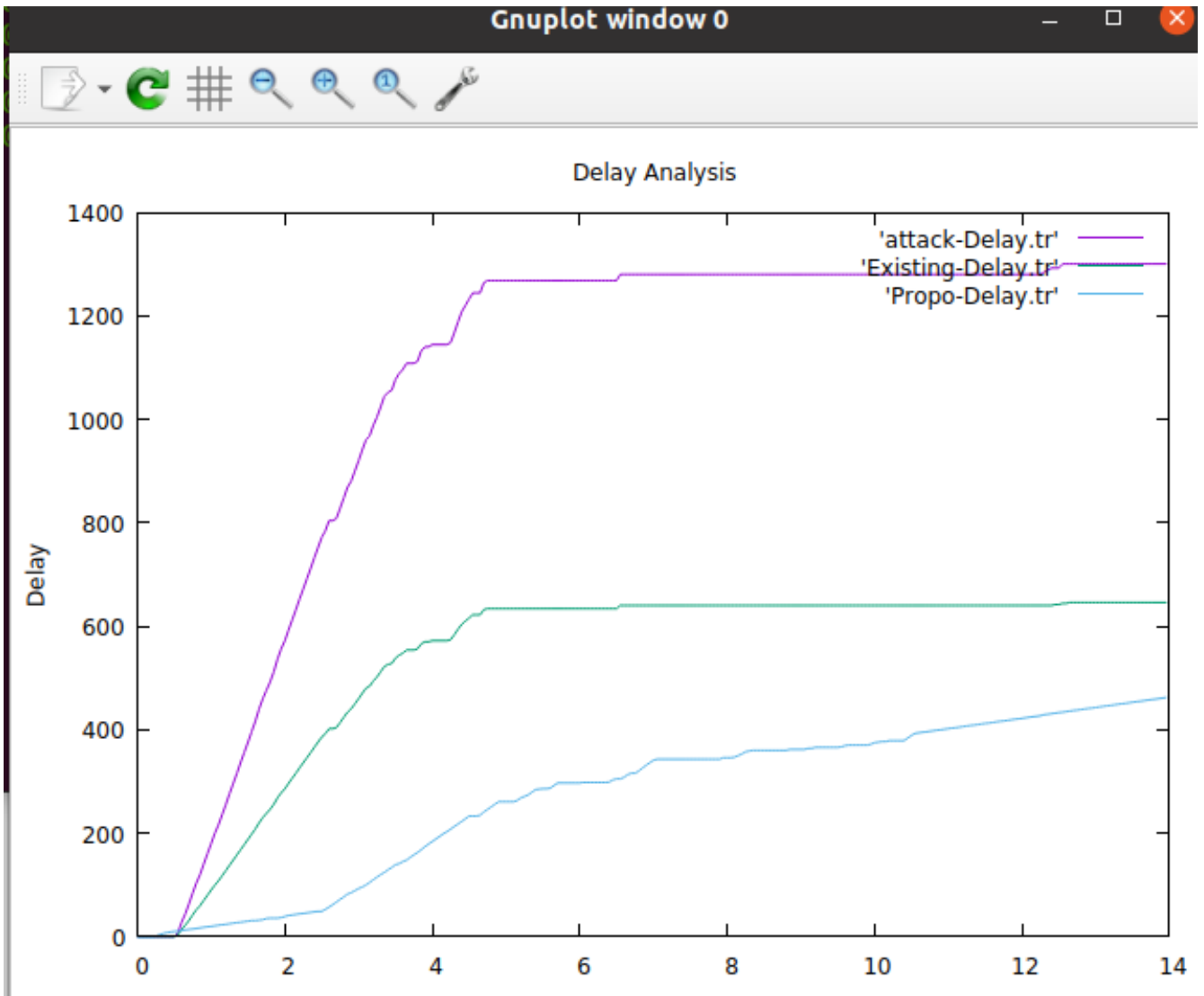


Figure 5.3: Delay analysis

Overhead of control message , as depicted in Fig 5.4, is the time it takes to accomplish a specific job of indirect computing using memory, bandwidth, or other resources. The attack model, the shield technique model, and the trust based

methods are all compared. As a tactic for separating the version number attack, the shield scenario is evaluated. The proposed scenario involves using a trust-based mechanism to isolate a version number attack.

The proposed scenario features fewer overhead control messages than other strategies for removing the network's version number assault.

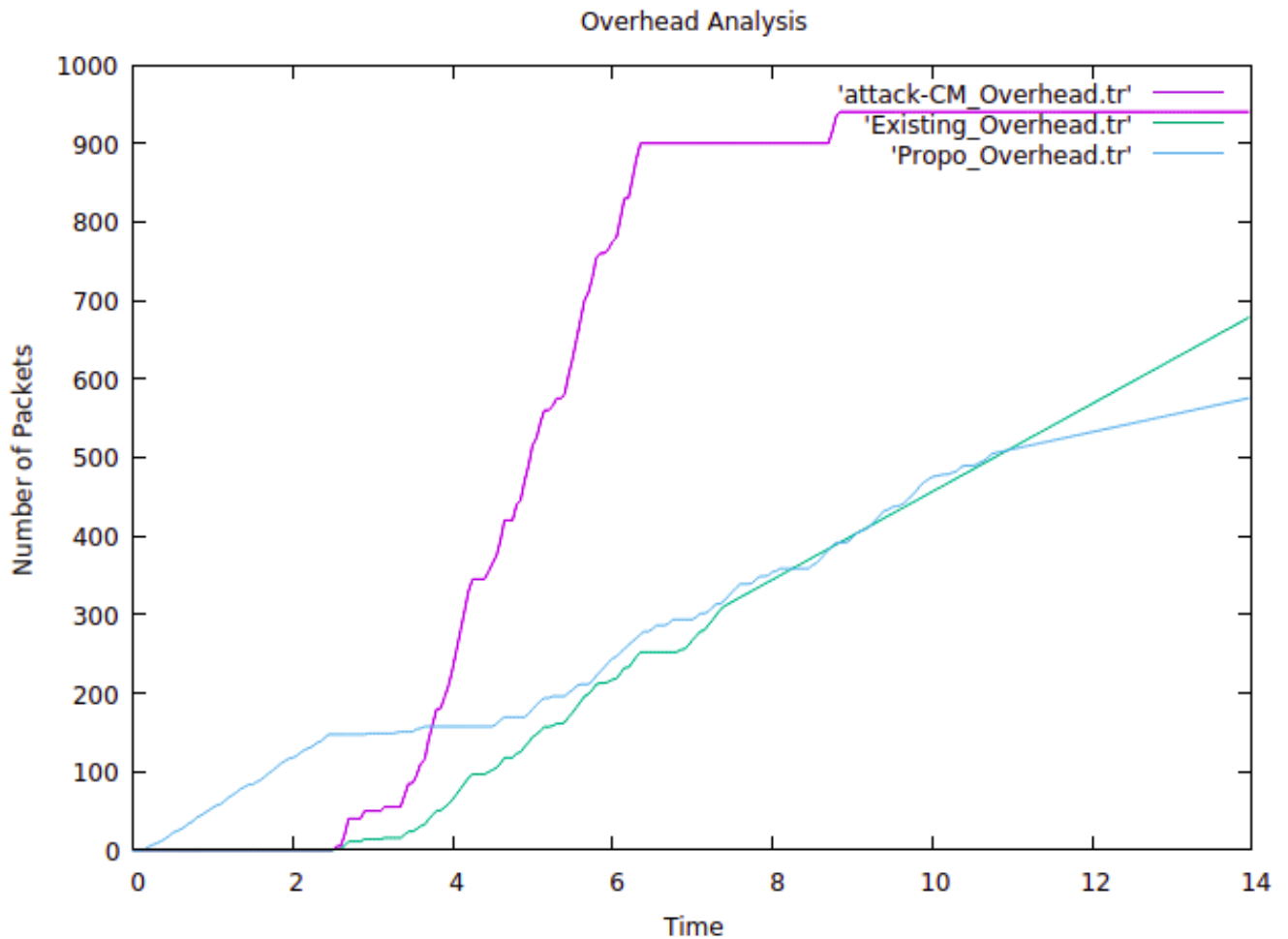


Figure 5.4: Overhead analysis

Because each sensor network node is dependent on the battery node, Fig 5.5 shows how the lifespan of a wireless sensor network. The proposed model utilises the least average power when compared to other strategies for VNA isolation in the network. Data packets may never arrive in the network if the network is congested and traffic monitoring is good.

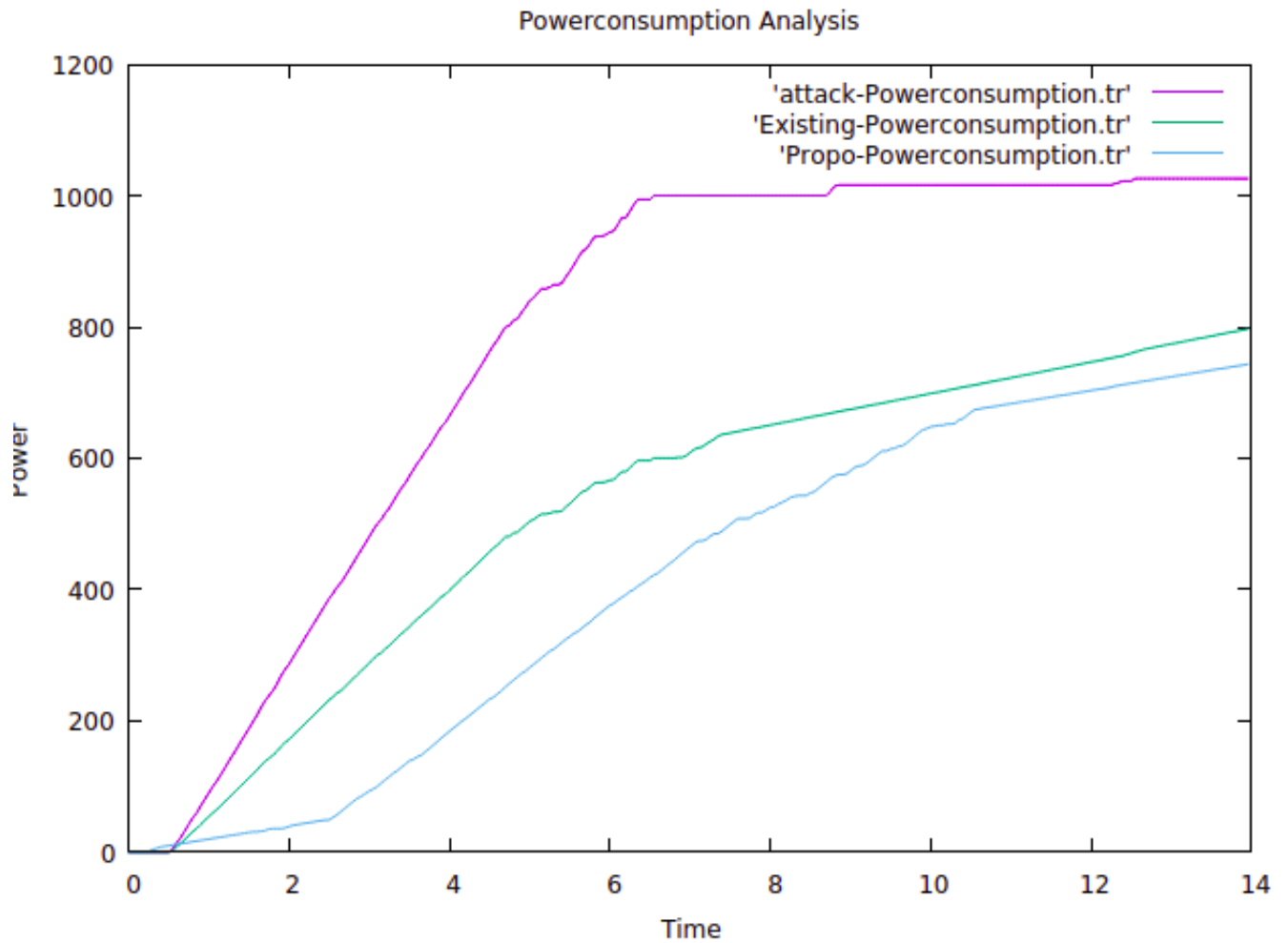


Figure 5.5: Power consumption analysis

Individual packets will never contribute to throughput and will have no impact on packet loss because they were never sent over the network. As shown in fig 5.6, the violet line, which represents the attack scenario, it is observed that the packet loss for the same gets increases, and it becomes constant when the attack is triggered. When the proposed technique is applied for the same, Packet loss is not increasing at a very high rate.

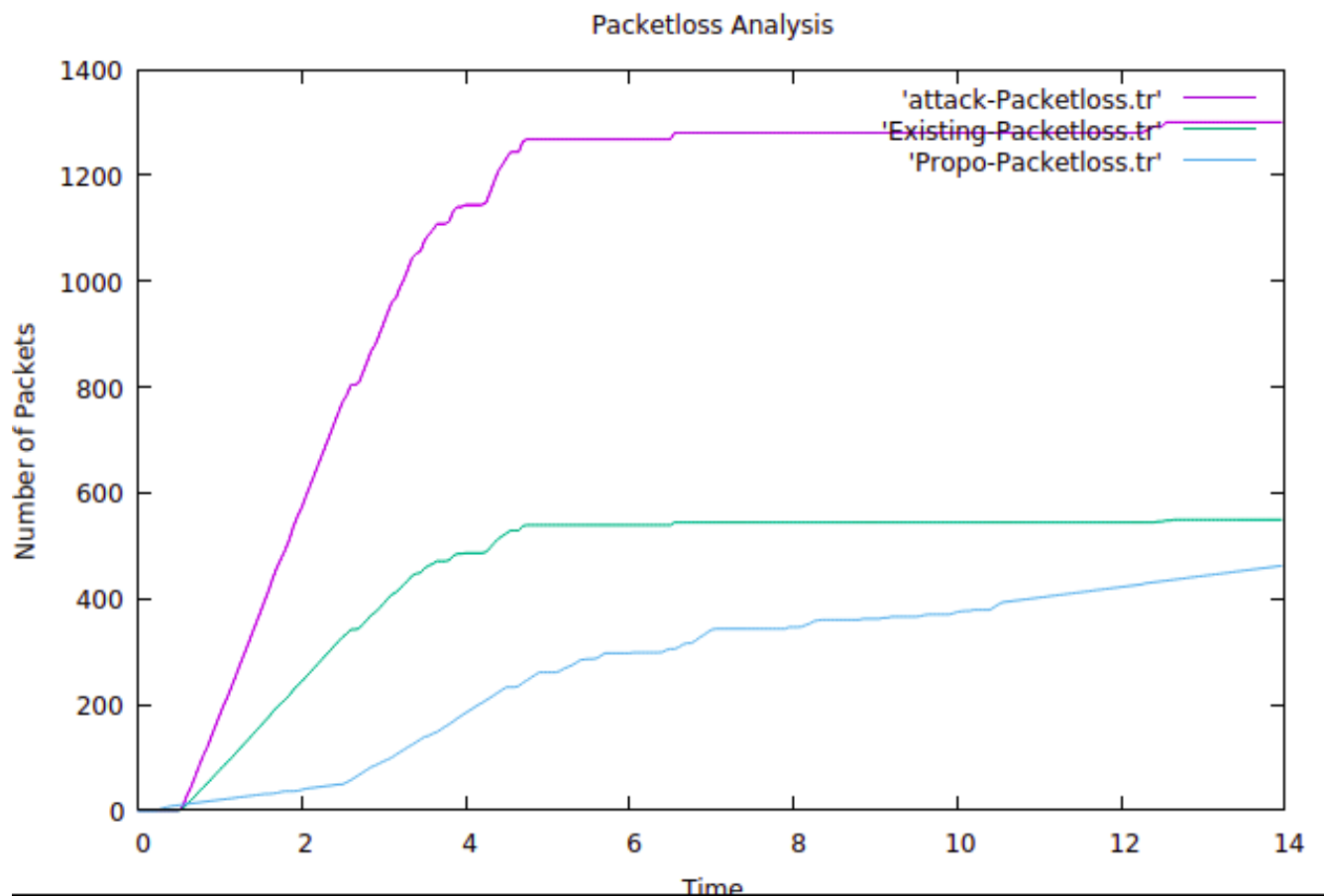


Figure 5.6: Packet loss analysis

Simulation for different numbers of tracker nodes is also carried out to see its effect on the network. The below table 5.1 describes the change in the parameter as we change the number of attacker nodes.

Attacker Node	One	Two
Delay	268.5571429	270.9964286
Throughput	109.4964286	91.225
Overhead	208.3910714	306.3146429
Power Consumption	397.7357143	404.3714286
PacketLoss	370.5558286	268.5571429

Table 5.1: Network performance with different attackers

As we can observe from Table 5.1 that proposed algorithm is capable of detecting the two malicious nodes from the deployed network topology. It can be clearly observed that the average throughput for two attacker node topology is very high and can results in significant loss in network performance. Removal of such nodes

from the topology is extremely important. As for the other parameters two attackers topology simulation shows higher values but with not much difference. In order to assess and comprehend the impact of the various node counts in the topology, Extensive simulation was carried out with a different number of nodes and their effect on the average of each performance metric when in attack mode. The next subsections will present simulation results based on the number of nodes for each performance indicator. Initially, the simulation was carried out when there were 38 nodes in the topology post.

No of nodes	38 nodes	30 nodes	20 nodes
Delay	1067.628	428.557	289.128
Throughput	48.132	36.125	25.521
Overhead	2607.41	923.178	685
Power Consumption	3776.778	583.55	418.564
PacketLoss	53.813	214.278	144.564

Table 5.2: Average of network performance parameters

The average of each performance metric was analyzed with 30 and 20 nodes, respectively. As we can analyze from the table 5.2, throughput, delay, and power consumption decrease as we reduce the number of nodes. This is primarily because as we decrease the total amount of packet transmissions, the amount of nodes in the network is also reduced also, Less of the number of energy will be required, so less power consumption in the topology. In the case of overhead and packet loss, topology does not behave as above because both the parameters are dependent on the number of packets transmitted and the position of the attacker node in the topology. As the position and number of attacker nodes change, the average of each parameter also changes.

No of nodes	50	70	90	110
Delay	239.944	354.225	371.132	392.7872
Throughput	67.373	121.65	148.61	121.584
Overhead	108.45	281.594	246.125	288.331
Power Consumption	338.538	513.97	497.263	551.265
PacketLoss	239.944	354.225	371.132	352.757

Table 5.3: Average of network performance parameters by increasing nodes

In order to study and comprehend the impact of various node counts in a topology. Extensive simulation was carried out further with a different number of nodes

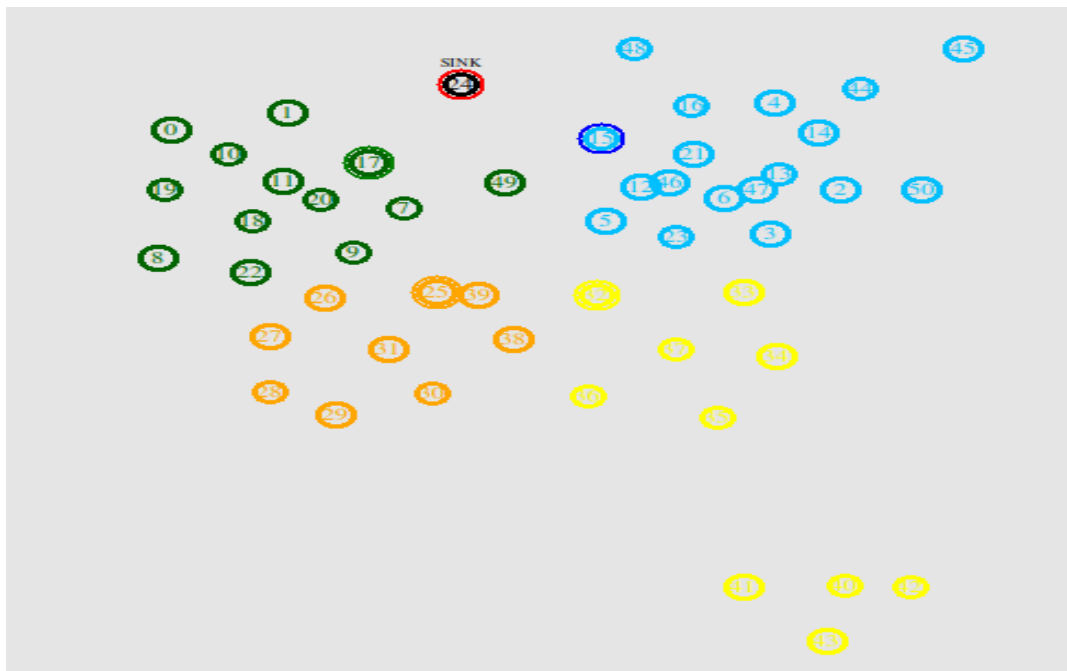


Figure 5.7: Topology with 50 nodes

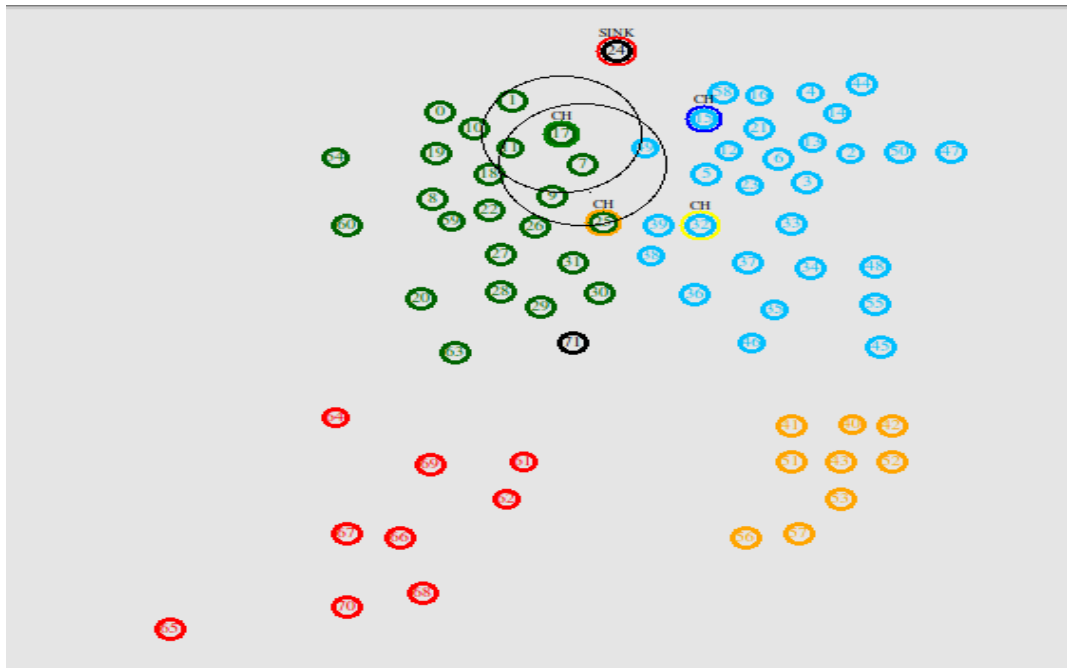


Figure 5.8: Topology with 70 nodes

and their effect on the average of each performance metric throughout timestamp. Following are the simulation results based on the number of nodes for each performance indicator. Initially, the simulation was carried out when there were 50 nodes in the topology. The average of each performance metric was analyzed with 70 and 90 nodes and 110 nodes, respectively.

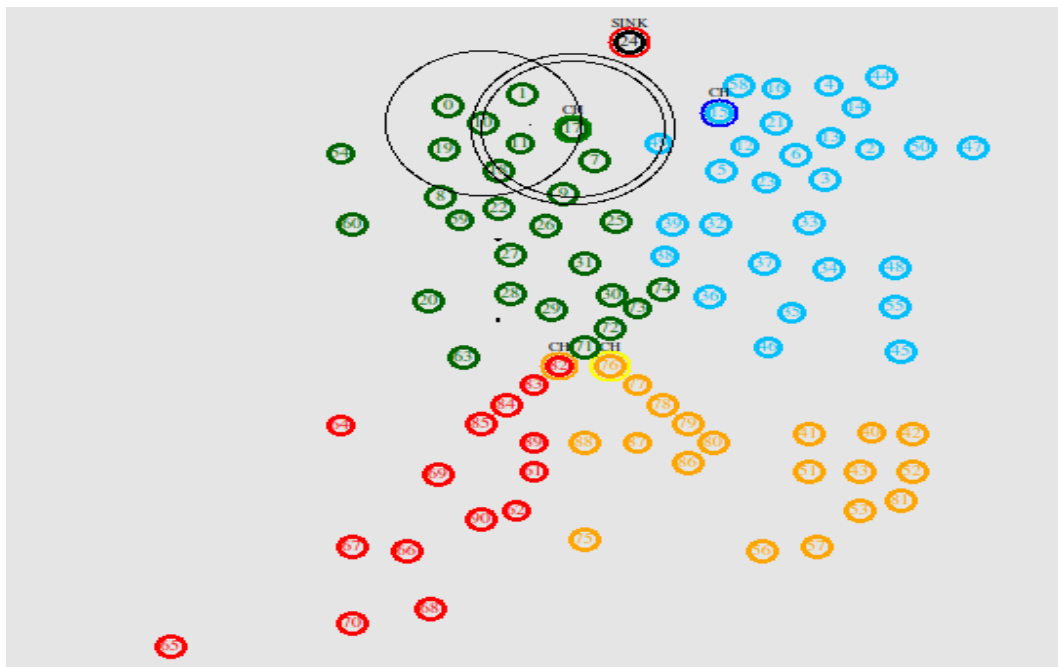


Figure 5.9: Topology with 90 nodes

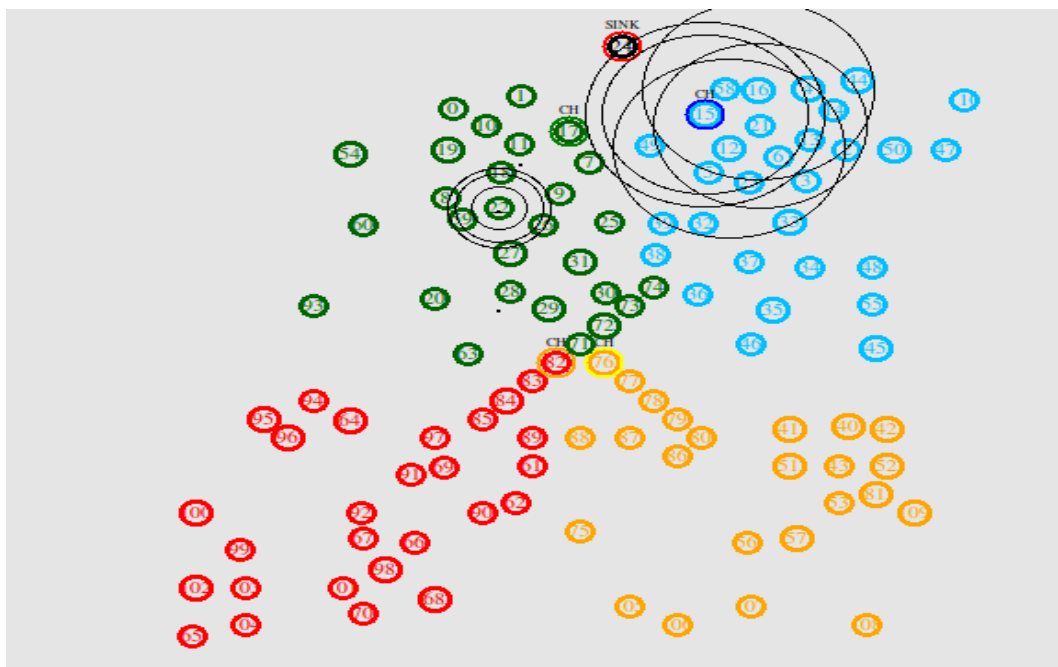


Figure 5.10: Topology with 110 nodes

As we can analyze from the table 5.3, packet loss, delay and overhead increase as we increase the quantity of all nodes. Congestion in network topology also grow, which results with more delay the quantity of nodes increases in the network. Since the number of packets is growing, packet loss and overhead are bound to in-

crease. More of the energy may be required, As a result, the topology's power consumption has increased. In case of throughput and power consumption, topology does not behave as above because both the parameters are dependent on the number of packets transmitted and the attacker node's position in the topology. Energy consumption and overhead do not show much difference as compared to other parameters.

5.2 Comparison with Existing Methods

VeRA (Version Number and Rank Authentication in RPL) method aims to provide security of rank and version number parameters by using one-way hash chain and message authentication codes.[5] However, the authors do not provide performance analysis in terms of memory, CPU, time and power. However, Power consumption is part of performance analysis in proposed work.

Mayzaud et al [14] analyzed the effects of version number attack in relation to the attacking positions on the grid topology. All of the nodes in their topology are static and it does not have any segments where nodes are randomly deployed with different densities. The performance results were provided in terms of control packet overhead, packet delivery ratio, average end-to-end delay, inconsistencies and loops. However, their work does not analyze the power consumption. It also does not accommodate any probabilistic attacking model. The simulation topology consists of 20 nodes in the grid topology. Attacker node relocates within the grid. Simulation is based on cooja and total simulation time is 50 minutes. Attack is triggered after every 5 minutes. It was observed that control packet overhead is 18 times as that of without attack scenario in some attacker position within grid and is maximum for the bottom row. Not only the neighbourhood of the attacker is impacted, but also the entire network. Proposed method consists of nodes ranging from 50 to 110 and is based on network simulator 2, and total simulation time is more than 60 minutes. Proposed method assesses the network better as attacker nodes are random placed and nodes are arranged in random topology. Mayzaud et al's [14] work lacks the extension to more complex topologies which is fulfilled in proposed model.

Ahmet [1] analyzed the version number attacks in a realistic scenario according to various points of view. In order to simulate a realistic scenario and understand how LLNs are expected to take place in application areas like industrial, building and home environments. Topology consists of 44 nodes, where 4 nodes are mobile and the rest are static. One node is the border router and is the root

of the DODAG. The simulations show that the packet delivery ratio and the control packet overhead results are highly correlated to the location of the attacker. In the worst case, the average power consumption of the nodes increases by 265 percent and almost reduces the average lifetime of the nodes to 14 of the usual. In terms of the power consumption and the average delay, the location of the attacker is not the major factor in the success of the attack. But technique proposed in this research work is entirely based on behavior of DIO messages arrival in order to detect the possible location of the version number attack while the proposed method is based on number of packets forwarded and received in the network at a particular node.

Detection strategy proposed by anthea mayzaud is based on a distributed monitoring architecture with dedicated algorithms that is able to identify malicious nodes performing such attacks in RPL based environments.[3] The performance of this solution is evaluated through extensive experiments and its scalability is quantified considering a monitoring node placement method. This approach has exploited monitoring nodes collaboration to identify the attacker, the attacker localization process being performed by the root after gathering detection information from all monitoring nodes. But exploitation of monitoring nodes in the network causes extra power consumption. Since devices in RPL are already resource constrained, renouncing the power consumption which plays a major role in performance analysis does not look promising.

5.3 Chapter Summary

Chapter 5 analyzes the results from the simulation and its effects on changing the topology, the number of attacker nodes, and a different number of nodes in the topology. Simulation results for all network performance parameters are plotted for attack scenario, shield and proposed technique. The end of subsection one presents the snapshot of topology with different number of nodes. Comparison between the various existing methods and proposed method is carried out in subsection two.

CHAPTER 6

Conclusion

This study aims at preventing version number attacks on the internet of things (IoT). The DODAG is a hierarchical architecture used in RPL for small devices in which malicious nodes increase the version number, resulting in the creation of a network link with a loop. It can lead to unauthorised global repair mechanism. Since RPL devices are resource constrained, this can impact the network lifetime as power consumption may increase.

Each DODAG is assigned a version number. The purpose of the version number is to ensure that there are loop free paths to the root node, the routing table entries of nodes in the DODAG are not obsolete and there is no inconsistency in the DODAG. The root node in a DODAG increments the version number in case of any inconsistency. This calls for a global repair process and the DAG is reconstructed. A malicious node may advertise a false version number in its control message to force a global repair.

This research proposes an optimized technique for mitigating network version number attacks and detecting malicious nodes. The trust-based technique will use the least amount of network resources. Simulation was carried out with different numbers of nodes in the network and results were analyzed in terms of network performance parameters. The proposed method is able to detect two version number attack nodes from the topology. Simulation was conducted with different number of nodes in topology to analyze its impact on performance parameters. Position of the attacker node also impacts its performance. Static nodes were used for simulation purpose. It was analyzed from the simulation that it affects the network to a great extent when the attacker is near the root node because this way it won't allow the message to reach root. Incorporation of mobile nodes can be done in the future work.

Chapter 6 concludes the research work.

References

- [1] J. Aris, S. F. Oktug, and S. B. O. Yalcin. Rpl version number attacks: In-depth study 2016 *ieee/ifip network operations and management symposium (noms 2016)*.
- [2] A. Aris, S. B. O. Yalc, and S. F. Oktu. New lightweight \checkmark mitigation techniques for rpl version number attacks, *ad hoc networks* (2018).
- [3] R. Badonnel and A. Mayzaud. Detecting version number attacks in rpl-based networks using a distributed monitoring architecture.
- [4] U. Bekcibasi and M. Tenruh. "increasing rssi localization accuracy with distance reference anchor in wireless sensor networks", *springer*, vol. 11, no. 8, 2014.
- [5] A. Dvir, L. Buttyan, and T. Holczer. Vera - version number and rank authentication in rpl.
- [6] D.Yin, L.Zhang, , and K.Yang. A ddos attack detection and mitigation with software-defined internet of things framework", *special section on security and trusted computing for industrial internet of things*, 2018.
- [7] H.Akram, Abdul-Ghani, and D. Konstantas. A comprehensive iot attacks survey based on a building-blocked reference model (*ijacsa*) *international journal of advanced computer science and applications*, vol. 9, no. 3, 2018.
- [8] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni. "bubbles of trust: A decentralized blockchain-based authentication system for iot", *computers security*, volume 78, september 2018, pages 126-142.
- [9] G. Han, L. Zhou, H. Wang, W. Zhang, and S. Chan. A source location protection protocol based on dynamic routing in wsns for the social internet of things", *future generation computer systems*, 2017.

- [10] S. Hud, J. Yearwood, M. M. Hassan, and A. Almogren. "securing the operations in scada-iot platform based industrial control system using ensemble of deep belief networks", *applied soft computing*, volume 71, october 2018, pages 66-77.
- [11] R. H. JHAVERI, N. M. PATEL, Y. ZHONG, and A. K. SANGAIAH. Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in industrial iot.
- [12] R. H. Jhaveri, N. M. Patel, Y. Zhong, and A. K. Sangaiah. Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in industrial iot", *special section on security and trusted computing for industrial internet of things*, 2018.
- [13] C.-T. Kuo, P.-W. Chi, V. Chang, and C.-L. Lei. Sfaas: Keeping an eye on iot fusion environment with security fusion as a service", *future generation computer systems*, volume 86, september 2018, pages 1424-1436.
- [14] A. Mayzaud, A. Sehgal, and R. Badonnell. Study of rpl dodag version attacks a. sperotto et al. (eds.): *Aims 2014, lncs 8508*, pp. 92–104, 2014. c ifip international federation for information processing 2014.
- [15] J. Moon and I. Y. Jung. "iot application protection against power analysis attack", *computers electrical engineering*, volume 67, april 2018, pages 566-578.
- [16] Y. Qian, Y. Jiang, J. Chen, Y. Zhang, J. Song, M. Zhou, and M. Pustišek. Towards decentralized iot security enhancement: A blockchain approach", *computers and electrical engineering* 72 (2018) 266–273.
- [17] R. H. Randhawa, A. Hameed, and A. N. Mian. "energy efficient cross-layer approach for object security of coap for iot devices", *ad hoc networks*, in press, corrected proof, available online 10 september 2018.
- [18] S. Rathore and J. H. Park. Semi-supervised learning based distributed attack detection framework for iot", *applied soft computing*, volume 72, november 2018, pages 79-89.
- [19] P. Sun, J. Li, M. Z. A. Bhuiyan, and B. L. Lihong Wang. "modeling and clustering attacker activities in iot through machine learning techniques", *information sciences*, in press, corrected proof, available online 23 april 2018.

- [20] B. Yigita, G. Gur, F. Alagoz, and B. Tellenbach. "cost-aware securing of iot systems using attack graphs", *ad hoc networks*, volume 86, 1 april 2019, pages 23-35.
- [21] Y.Liu, Y. Kuang, Y. Xiao, , and G. Xu. Sdn-based data transfer security for internet of things", *ieee internet of things journal*, 2018.

CHAPTER A

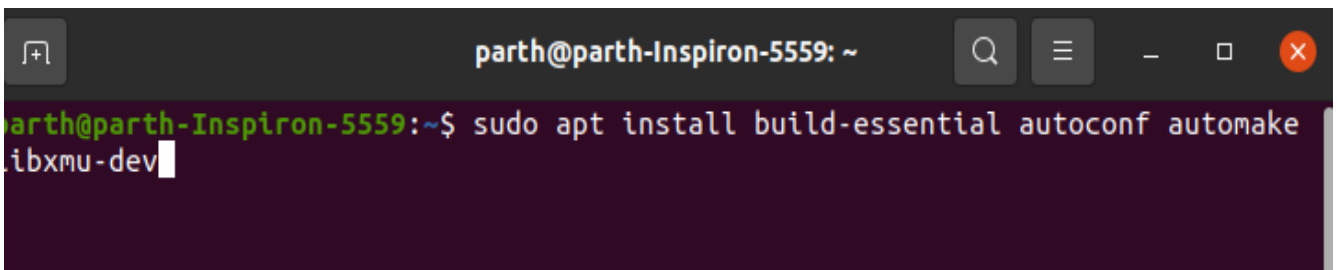
Installation and Setting up for Network Simulator 2

Version for ns-2 : ns-2.36

Ubuntu version : 20.04 (LTS)

Step A: Download and install essential libraries and packages of ubuntu required for installation.

"sudo apt install build-essential autoconf automake libxmu-dev"

A terminal window screenshot with a dark background. The title bar shows 'parth@parth-Inspiron-5559: ~'. The terminal prompt is 'parth@parth-Inspiron-5559:~\$' followed by the command 'sudo apt install build-essential autoconf automake libxmu-dev' with a cursor at the end of the line.

```
parth@parth-Inspiron-5559:~$ sudo apt install build-essential autoconf automake libxmu-dev
```

Figure A.1: Installing essential libraries

Step B: Next step is to install the latest versions of gcc (4.8) we need to open sources.list file in sudo mode

"sudo nano /etc/apt/sources.list"

Append the following piece of text in the file , close and save it.

"deb http://in.archive.ubuntu.com/ubuntu bionic main universe"

Step C: In a terminal, run "sudo apt update"

```
parth@parth-Inspiron-5559: ~
GNU nano 4.8 /etc/apt/sources.list
deb cdrom:[Ubuntu 20.04.3 LTS _Focal Fossa_ - Release amd64 (20210819)]/ focal
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://in.archive.ubuntu.com/ubuntu/ focal main restricted
# deb-src http://in.archive.ubuntu.com/ubuntu/ focal main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://in.archive.ubuntu.com/ubuntu/ focal-updates main restricted
# deb-src http://in.archive.ubuntu.com/ubuntu/ focal-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://in.archive.ubuntu.com/ubuntu/ focal universe
# deb-src http://in.archive.ubuntu.com/ubuntu/ focal universe
deb http://in.archive.ubuntu.com/ubuntu/ focal-updates universe
# deb-src http://in.archive.ubuntu.com/ubuntu/ focal-updates universe

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^I Paste Text ^T To Spell ^_ Go To Line
```

Figure A.2: Appending code to sources.list

```
parth@parth-Inspiron-5559:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 https://dl.google.com/linux/chrome/deb stable InRelease
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Hit:6 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata
ta [278 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Me
tadata [391 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11
Metadata [944 B]
Get:10 http://in.archive.ubuntu.com/ubuntu focal-backports/main amd64 DEP-11 Met
adata [9,592 B]
Get:11 http://in.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11
Metadata [30.7 kB]
Fetched 932 kB in 2s (547 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
126 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Figure A.3: Update linux packages

Step D: In a terminal, run "sudo apt install gcc-4.8 g++-4.8"

Step E: Extract the ns2 files which were downloaded earlier.

The commands used for the same are listed below.

```
"tar zxvf ns-allinone-2.35.tar.gz"
```

```
"cd ns-allinone-2.35/ns-2.35"
```

Step F: Next we need to make changes in all make files in all subdirectories which are listed below.

1. nam-1.15/Makefile.in
2. xgraph-12.2/Makefile.in
3. otcl-1.14/Makefile.in

Make the following changes in each of the files.

```
"@CC@ -> gcc-4.8"
```

```
"@CXX@ -> g++-4.8"
```

Open following file directory:

```
" ns-2.35/linkstate/ls.h"
```

Go to line 137 and change the following

```
"void eraseAll() erase(baseMap::begin()),"  
"baseMap::end());"
```

to this

```
"void eraseAll() this-> erase(baseMap::begin()),"  
"baseMap::end());"
```

This is an extra change that we have to make.

Step G: Get a new terminal open parallel.

Change the directory to ns-allinone-2.35/

Install the ns2 using the ./install command in the terminal.

Step H - Now we need to change the environment variable path

Get a new terminal open parallel.

Run the command gedit .bashrc

The image shows two side-by-side terminal windows. The left window displays the output of an `apt update` command, showing progress for various Ubuntu repositories like focal-updates, focal-backports, bionic, and universe. The right window shows the contents of the `~/.bashrc` file, which includes comments and configuration for the bash shell, such as setting the `PATH` and `LD_LIBRARY_PATH` variables.

```
parth@parth-Inspiron-5559: ~  
et:4 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
et:5 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]  
et:6 http://in.archive.ubuntu.com/ubuntu bionic InRelease  
et:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [278 kB]  
et:8 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [391 kB]  
et:9 http://in.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]  
et:10 http://in.archive.ubuntu.com/ubuntu focal-backports/main amd64 DEP-11 Metadata [9,592 B]  
et:11 http://in.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [30.7 kB]  
etched 932 kB in 2s (547 kB/s)  
  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
26 packages can be upgraded. Run 'apt list --upgradable' to see them.  
parth@parth-Inspiron-5559:~$ gedit ~/.bashrc  
  
1 # ~/.bashrc: executed by bash(1) for non-login shells.  
2 # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)  
3 # for examples  
4 export PATH=$PATH:/home/parth/ns-allinone-2.35/bin:/home/parth/ns-allinone-2.35/tcl8.5.10/unix:/home/parth/ns-allinone-2.35/tk8.5.10/unix  
5 export LD_LIBRARY_PATH=/home/parth/ns-allinone-2.35/otcl-1.14:/home/parth/ns-allinone-2.35/lib  
6 # If not running interactively, don't do anything  
7 case $- in  
8   *(*) ;;  
9   *) return;;  
10 esac  
11  
12 # don't put duplicate lines or lines starting with space in the history.  
13 # See bash(1) for more options  
14 HISTCONTROL=ignoreboth  
15  
16 # append to the history file, don't overwrite it
```

Figure A.4: Setting path using `~/.bashrc`

Add the `PATH` and change the `LDLIBRABRYPATH`

Step I: In order to make the changes visible , Restart the computer.

Step J: end