

Analysing the impact of packet dropping attack in IOT

Project report submitted in partial fulfilment of the requirement for the
degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Manan Sardana (191535)
V.Saicharan (191530)
Pradhyum Bhati (191538)

Under the supervision of

Mr. Arvind Kumar

to



Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology Waknaghat, Solan-
173234, Himachal Pradesh**

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled “**Analysing the impact of packet dropping attack in IOT**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of **(Mr.Arvind Kumar) (Assistant professor grade II)**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)

Manan Sardana (191535)

Pradhyum Bhatti (191538)

V Saicharan (191530)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Supervisor Name- Mr. Arvind Kumar

Assistant Professor (grade II)

Computer Science & Engineering and Information Technology

Dated:

Acknowledgement

Firstly, We express our heartiest thanks and gratefulness to almighty God for His divine blessing making it possible to complete the project work successfully.

We are really grateful and wish our profound indebtedness to Supervisor **Mr. Arvind Kumar , Assistant Professor (Grade II)**, Department of Computer Science and Engineering, Jaypee University of Information Technology, Solan Deep knowledge & keen interest of my supervisor in the field of “**IOT**” helped us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would also generously welcome each one of those individuals who have helped us straightforwardly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

Manan Sardana

[191535]

V.Saicharan

[191530]

Pradhyum Bhati

[191538]

Table Of Contents

Title	Page No.
Chapter-1 Introduction	1-14
Chapter-2 Literature Survey	15-22
Chapter-3 System Development	23-46
Chapter-4 Performance Analysis	47-56
Chapter-5 Conclusion	57-60
References	61-62

LIST OF FIGURES

Figure 1.2.1 RPL DODAG instances

Figure 1.2.2.3a Packet drop in RPL network

Figure 1.2.2.3b Packet drop attack

Figure 1.3a shows cooja in the ubuntu os.

Figure 1.3 Cooja Simulator with modes arranged.

Figure 1.3b Collect Data view in Cooja

Figure 3.1a Cooja Simulator flow diagram

Figure 3.1b Contiki OS

Figure 3.1b Cooja Simulator UI.

Figure 3.1c Virtual Machine with cooja Simulator.

14)

Figure 3.2.1a Install Cotinki Instant.

Figure 3.2.1b Extracted and Zip downloaded file

Figure 3.2.1c Cooja disk file.

Figure 3.2.1d Starting the virtual machine of Cooja

Figure 3.2.1e Password field in VM of Coojauser

Figure 3.2.1f Cotinki screen

Figure 3.2.2a Cooja initial screen

Figure 3.2.2b Creating new stimulation.

Figure 3.2.2c Important cooja settings

Figure 3.2.2d Adding skymode.

Figure 3.2.2e Selecting the udp-sink.c file.

Figure 3.2.2f Adding the created mode node

Figure 3.2.2f Added the created mode node.

Figure 3.2.2g Adding the created mode node.

Figure 3.2.2h No Packet attack structure.

Figure 3.2.2i Collect view in cooja

Figure 3.2.2j Collect View screen

Figure 3.2.2k Starting simulation by start button on Simulation Control panel.

Figure 3.2.3a Decreased Packet drop attack folder created.

Figure 3.2.3b Changes in the code of rpl-private.h

Figure 3.2.3c Changes in the code of rpl-timers.c

Figure 3.2.3e Decreased Packet drop attack data collection structure

Figure 3.2.4a Increased Packet drop attack folder created.

Figure 3.2.4c Changes in the code of rpl-iemp6.c

Figure 4.1.1a Node information of no attack stimulation.

Figure 4.1.1b Node information of decreased Packet drop attack stimulation.

Figure 4.1.1c Sensor map of no Packet drop attack stimulation.

Figure 4.1.Id Sensor map of decreased Packet drop attack stimulation.

Figure 4.1.2a and c Average Power Consumption Graph of no rank and Packet drop attack stimulation.

Abstract

This study plans to examine the effect of bundle dropping assaults in IoT networks utilizing the COOJA test system. The review centres around the effect of such goes after on the dependability and execution of IoT organizations, explicitly on the transmission of information between IoT gadgets and a door. The reproduction includes making an organization geography with different IoT gadgets and a passage, and mimicking a parcel dropping assault on the organization.

The aftereffects of the reproduction show that bundle dropping assaults can essentially affect the dependability and execution of IoT organizations, prompting postponements and parcel misfortune. The concentrate likewise explores the effect of different assault boundaries like assault term and assault force on the organization execution. The discoveries propose that more extended assault terms and higher assault powers can bring about additional huge interruptions to the organization.

The review reasons that parcel dropping assaults can have extreme results on the dependability and execution of IoT organizations, prompting disturbances and postpones in information transmission. The recreation results can be utilized to foster compelling methodologies and measures to relieve the effect of such goes after in genuine IoT organizations. By and large, this study gives bits of knowledge into the effect of bundle dropping assaults in IoT organizations and features the significance of creating compelling safety efforts to shield IoT networks from such assaults.

Chapter-1 Introduction

1.1 Introduction

The Internet of Things (IoT) is a rapidly growing field with a wide range of applications, including smart homes, industrial automation, and healthcare. One of the fundamental challenges in IoT is collecting data from a large number of interconnected devices. To address this issue, researchers have proposed various data collection algorithms, including tree-based, cluster-based, and geographic-based approaches.

However, the security of these data collection algorithms is a critical concern, as IoT devices are vulnerable to various cyber attacks. One such attack is the packet dropping attack, in which an attacker selectively drops packets to disrupt the data collection process. This attack can cause significant damage, including the loss of valuable data and potential system failures.

In this report, we will analyze the impact of packet dropping attacks on data collection in IoT using the COOJA simulator. COOJA is a popular tool for simulating IoT networks and evaluating the performance of data collection algorithms. By simulating various scenarios with different attack intensities, we will investigate the impact of packet dropping attacks on different data collection algorithms.

The report is organized as follows. In section 3, we will provide a brief overview of packet dropping attacks and the COOJA simulator. It will describe the methodology used to evaluate the impact of packet dropping attacks on data collection in IoT. Section 4 will present the results of our experiments, followed by a discussion of the findings. Finally, section 5 will conclude the report and provide recommendations for improving the security of data collection in IoT.

1.1.1 RPL Routing Protocol Overview

Routing protocols are an essential component of wireless sensor networks (WSNs), including those used in Internet of Things (IoT) systems. One common routing protocol used in WSNs is the Reactive Protocol (RPL), which is designed specifically for low-power and lossy networks.

RPL builds a directed acyclic graph (DAG) to represent the topology of the network, with nodes serving as both routers and destinations for data packets. The DAG provides a route from a source node to a destination node, and RPL uses a combination of proactive and reactive mechanisms to establish and maintain this route.

In terms of packet dropping attacks, these can have a significant impact on data collection in IoT systems. Packet dropping attacks involve an attacker dropping packets at specific nodes in the network, which can lead to data loss, delay, or corruption.

To analyze the impact of packet dropping attacks on data collection in IoT using COOJA, a network simulation tool, it is essential to simulate the attack and measure its effects on the network's performance. This can be done by configuring a network in COOJA with a RPL routing protocol and then simulating the packet dropping attack under different scenarios.

1.1.2. RPL Security

Due to characteristics that are inherited from other networks, secure routing methods for data packets in IoT networks are a hard challenge [6]. Because it involves users, possible security threats to data packet routing in IoT-restricted devices have a significant impact. Several RPL attacks take place due to the actions of hostile nodes while data packets are being routed between devices [8]. In [9-11], the RPL security was thoroughly examined. Numerous RPL attacks have been examined, although the majority of the studies have

not focused on secure RPL's processes. For IoT-restricted devices, efforts have recently been undertaken to build safe routing protocols. They all, however, rely on traditional cryptographic operations, which severely deplete device resources and have an adverse effect on the performance of the restricted devices that are most likely to be employed in IoT applications. Due to their lack of infrastructure, unreliable links, resource limitations, poor physical security, and changing topology, RPLs are weak points that are challenging to defend against intrusions. The high volume of data traffic in some nodes can cause them to run out of power and resources more quickly than other nodes, deactivating the root node's data routing. The network has billions of connected devices, making it extremely difficult to secure and protect them from many types of threats. Users would believe their data is not safe if these devices are susceptible to assaults like rank and version number attacks .

1.2 Problem Statement

Packet dropping attacks are a type of cyber attack that can significantly impact the performance of IoT networks. These attacks involve dropping packets at specific nodes in the network, which can lead to data loss, delay, or corruption. As IoT networks become increasingly common, it is essential to understand the impact of packet dropping attacks on data collection in these networks.

To address this problem, it is necessary to analyze the impact of packet dropping attacks in IoT networks using COOJA, a network simulation tool. By simulating these attacks under different scenarios and configurations, it is possible to gain a better understanding of their effects on the performance of IoT networks. Specifically, this research aims to analyze the impact of packet dropping attacks on data collection in IoT networks that use the RPL routing protocol, which is commonly used in low-power and lossy networks.

The analysis of the impact of packet dropping attacks in IoT networks will contribute to the development of strategies for preventing or mitigating these attacks. Additionally, this research will provide insights into the security of IoT networks and inform the development of more robust and resilient systems.

1.2.1. RPL DODAG

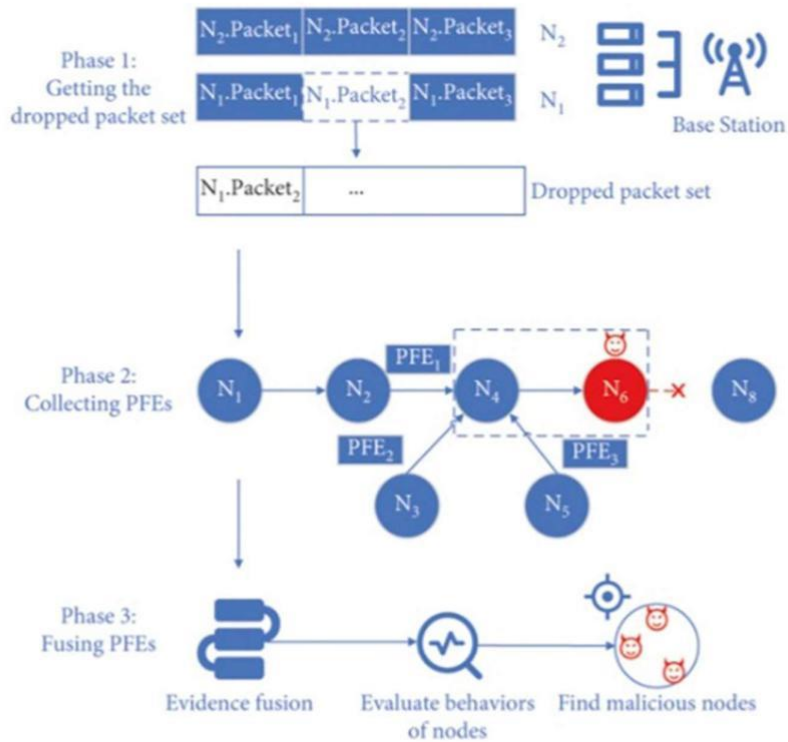
In the context of analyzing the impact of packet dropping attacks on data collection in IoT using COOJA, the RPL (Routing Protocol for Low-Power and Lossy Networks) DODAG (Destination-Oriented Directed Acyclic Graph) plays a crucial role.

The RPL protocol is commonly used in IoT networks and is designed for low-power and lossy networks. The protocol creates a directed acyclic graph (DAG) that is used to route packets between nodes in the network. The DAG consists of nodes organized into a hierarchy, with a single root node at the top and leaf nodes at the bottom. The DAG is constructed using a DODAG, which is a DAG with a specific root node and a specific set of edges.

In COOJA, it is possible to simulate the impact of packet dropping attacks on data collection in IoT networks that use the RPL protocol and its DODAG structure. By simulating packet dropping attacks at specific nodes in the DODAG, it is possible to measure the impact of these attacks on network performance and data collection.

For example, a study by S. Gautam and N. Kumar, "An Experimental Analysis of Packet Dropping Attacks in IoT-Based Wireless Sensor Networks," used COOJA simulations to analyze the impact of packet dropping attacks on IoT-based WSNs that use the RPL protocol and its DODAG structure. The study found that packet dropping attacks can significantly impact the performance of the RPL protocol, resulting in a decrease in packet delivery ratio and an increase in end-to-end delay. The study also analyzed the effectiveness of different mitigation strategies, such as modifying the RPL parameters and implementing security measures, in reducing the impact of packet dropping attacks.

Overall, the RPL DODAG is a critical component of analyzing the impact of packet dropping attacks on data collection in IoT using COOJA. By simulating these attacks in the context of the RPL protocol and its DODAG structure, it is possible to gain a better understanding of their impact on network performance and develop effective strategies for mitigating their effects.



As mentioned above, EFDA does not need to inject extra packets to obtain the training dataset to train the detection model, and it utilizes the existing PFEs in the network to perform.

1.2.2. Different types of attacks:

1.2.2.1 DIS Attack

The flooding attack will be used against the IoT nodes in this attack. The flooding attack is a denial-of-service attack since it tries to disable nodes and links by producing a lot of traffic. This form of attack is dangerous as long as the goal is to disrupt the service and disable nearby nodes as well as the service overall [18]. The attacking node will cause other nodes to lose energy, experience latency, and lose packets.

1.2.2.2 DIO Flood Attack

This simulation aims to show the impact of the flood attacks, not only with DIS messages but also with DIO messages. The goal of a DIO Flood attack is to send a significant amount of DIO messages to the malicious node's neighbours in an effort to drain all of its internal resources by trying to serve the fake traffic so that it cannot handle any legitimate incoming service requests.

1.2.2.3 Version Number Attack

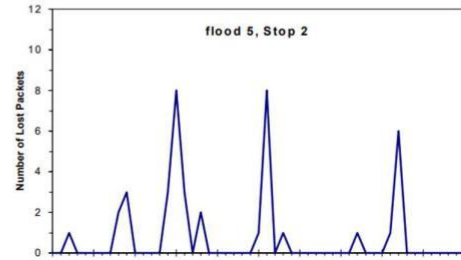
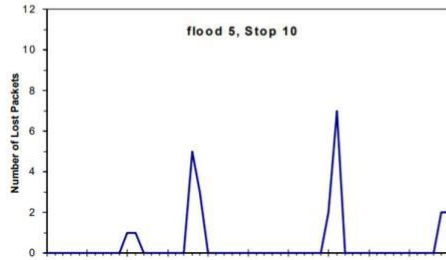
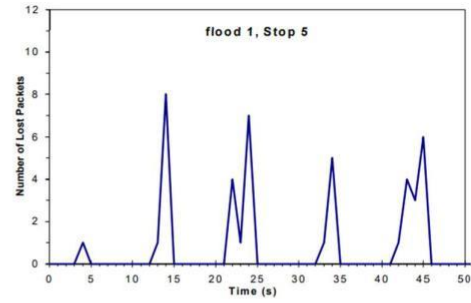
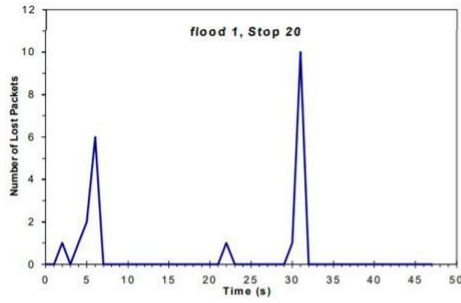
As long as the IoT devices are energy-constrained devices, this type of attack keeps sending a higher version of DODAG each time. This behaviour will force the IoT network topology to be reset, the DODAG tree to trigger another topology formation, and it can cause instability in the nodes' topology and further energy waste [19]. The goal of this simulation is to examine the impacts of this assault on energy usage, radio traffic, and data traffic when the IoT nodes periodically receive DIO messages with a higher version. A solution might also be suggested to stop or lessen this onslaught [20].

1.2.2.3 Black Hole Attack

A black hole attack's main goal is to launch a denial of service attack against the leaf nodes. In this case, the malicious node intercepts all control messages sent by the other nodes in the network and absorbs them. The network may become disturbed as a result of this. To create a valid DODAG without problems, the nodes should be able to interact effectively [21]. Additionally, the rogue node does not produce any control messages when executing a black hole attack. Some nodes may be cut off from the network as a result. We discussed how this assault affected the high packet loss rate, control, route traffic overhead, as well as node energy usage.

1.2.2.4 Packet dropping Attack

A packet dropping attack in IoT (Internet of Things) refers to a malicious activity where an attacker selectively drops or discards network packets that are transmitted between IoT devices or between an IoT device and a gateway or cloud server. This type of attack can result in communication disruptions, delay or even complete failure of the IoT network. Packet dropping attacks can be initiated by various means, including exploiting vulnerabilities in the IoT devices or network protocols, using rogue access points, or performing a distributed denial of service (DDoS) attack. Attackers can drop packets selectively to target specific IoT devices or disrupt specific types of traffic, such as critical system updates or sensitive data transfers. To mitigate packet dropping attacks in IoT, several measures can be taken. These include implementing secure communication protocols such as Transport Layer Security (TLS) or Secure Sockets Layer (SSL), implementing network traffic monitoring and intrusion detection systems, using packet filtering techniques, and regularly updating IoT devices with the latest security patches and firmware updates. It is also essential to conduct regular security assessments and penetration testing of IoT networks to identify vulnerabilities and potential attack vectors. In addition, educating end-users and IoT device manufacturers on the importance of security practices can help prevent packet dropping attacks and other security threats in the IoT ecosystem.



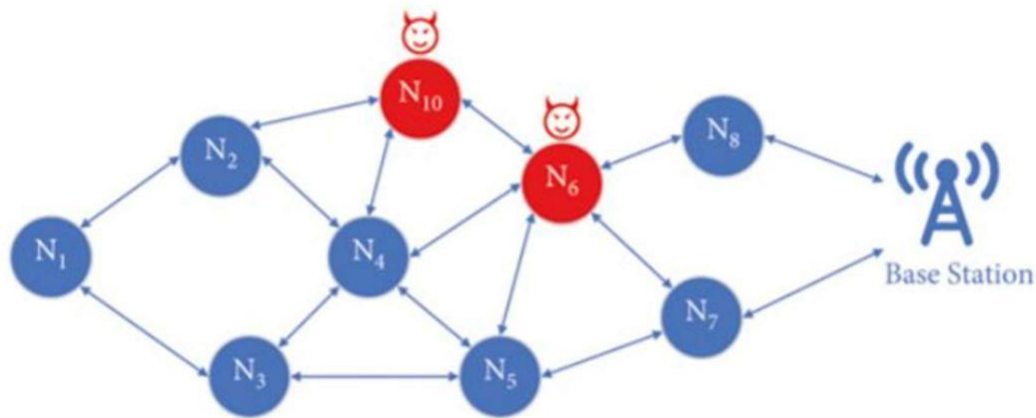
1.3 Objectives

The main objective that we have in this project is to implement the normal RPL topology and collect the data that we get and compare it with the attacked nodes topology and collect the data for it.

Here we will implement the no attack topology in the simulator cooja and then find the data and analysis that we get using the sink node and sender nodes in the collect view. Then we will implement the increased Packet drop attack topology in the simulator cooja and then find the data and analysis that we get using the sink node and sender nodes in the collect view. Then we will implement the decreased Packet drop attack topology in the simulator cooja and then find the data and analysis that we get using the sink node and sender nodes in the collect view.

After implementing the no attack and attack topology and running it in the simulator we compare the data we get from the three cases and compare all the graphs and information

we get and do the analysis to find an algorithm that helps us detect and solve the Packet drop attack issue.



A node is represented as, and the base station is represented as. Each node has at least one routing path to the base station. A routing path is represented as, which is expressed as where it represents a packet which is sent from forwarded through in a sequence, and finally received by the base station . Then, the network is expressed as

In the figure 1.3a we can see the simulator that we need to use to implement the attack and no attack. We need to open the simulator and run it for the tests of attacks and no attacks.

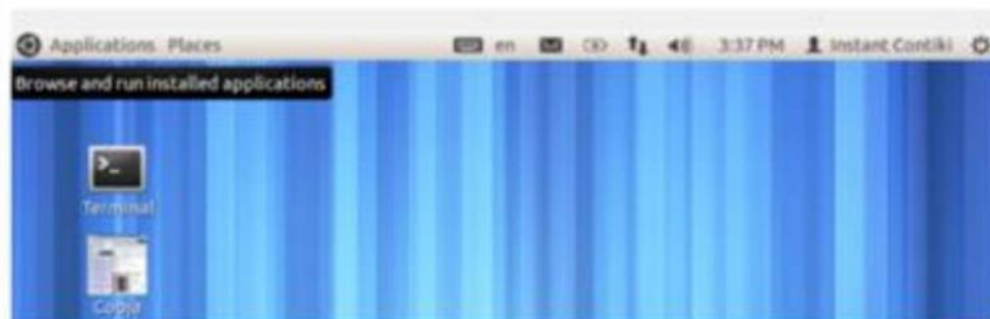


Figure 1.3a shows cooja in the ubuntu os.

In the figure 1.3b we can see the cooja simulator opened and created topology of nodes in the modes arrange section. The mode number 1 and colored green is the sink

mode and all other nodes in yellow colour are sender and receiver nodes which send and receive the data and packets from each other.

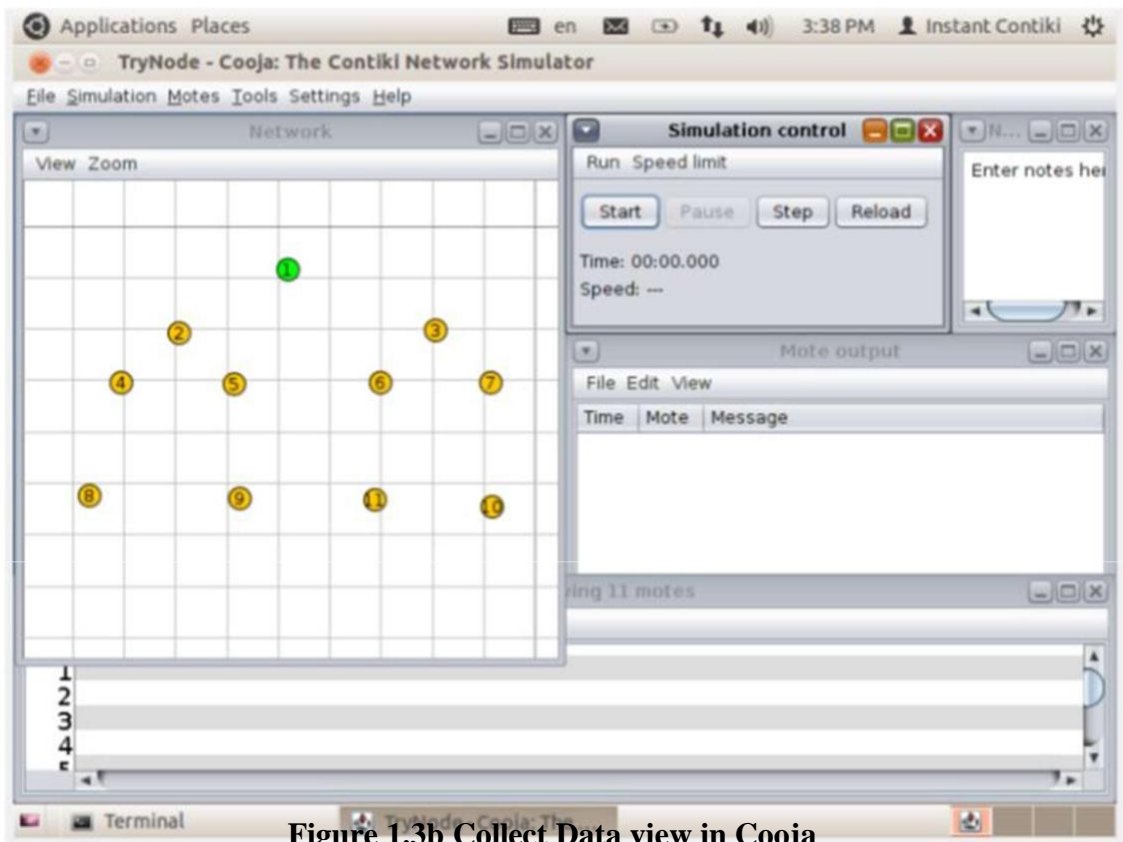


Figure 1.3b Collect Data view in Cooja

In the figure 1.3c we can see the data collect mode that is opened in the cooja simulator and is needed to analyse and collect the data we get from the all the no attacks and attacks implemented. So our major objective is to find the data from all the three no attack, decreased Packet drop attack and increased Packet drop attack. Compare them using the graphs and analysis and construct an algorithm to find a way to detect and solve both of the Packet drop attack problems.

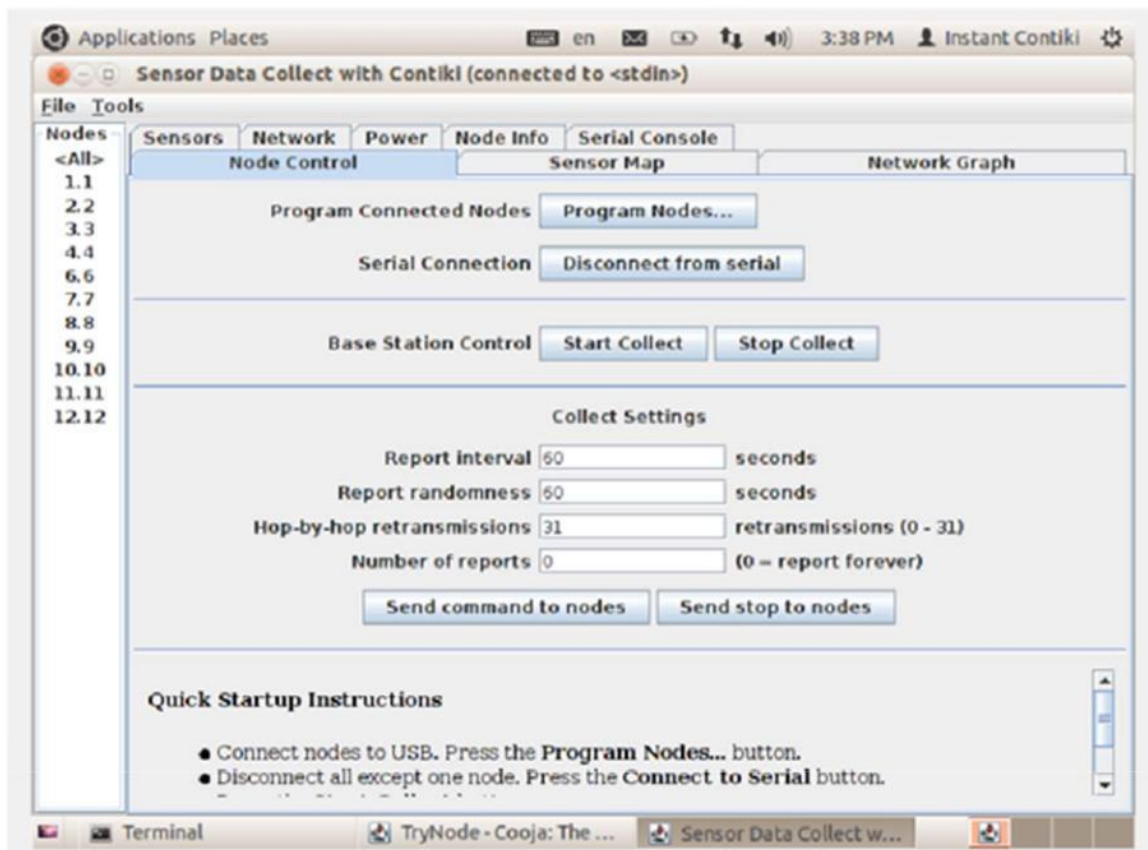
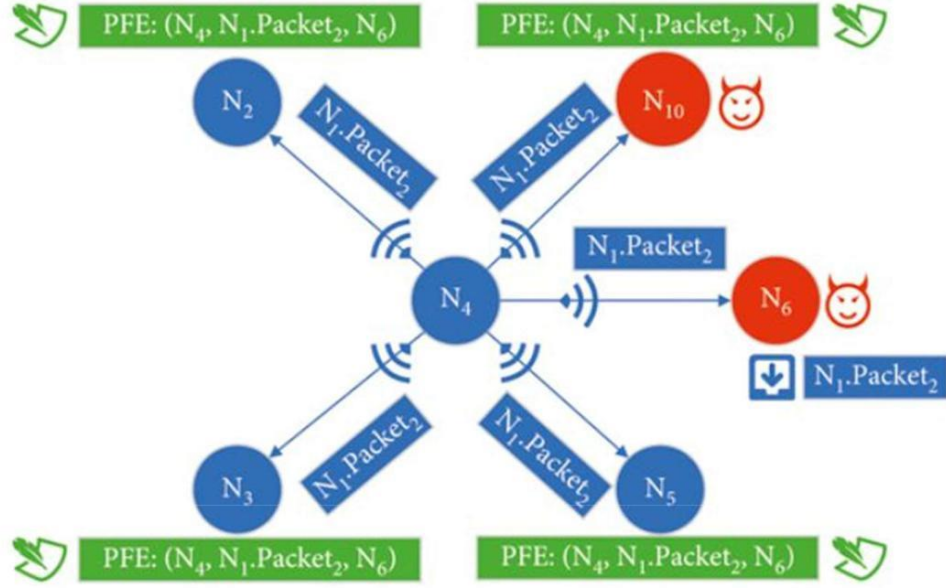


figure 1.3c data collect mode in cooja

1.4 Methodology

1. Network configuration: The first step is to configure the network in COOJA with the RPL routing protocol. This involves defining the number of nodes in the network, their locations, and their roles as either source or destination nodes.
2. Packet dropping attack simulation: The next step is to simulate the packet dropping attack under different scenarios. This involves dropping packets at specific nodes in the network and measuring the impact on data collection, such as the amount of data loss or delay.
3. Performance evaluation: After simulating the packet dropping attack, it is necessary to evaluate the performance of the network. This involves measuring metrics such as packet delivery ratio, end-to-end delay, and throughput to determine the impact of the attack on the network's performance.
4. Data analysis: The next step is to analyze the data collected from the simulation and performance evaluation. This involves identifying patterns or trends in the data, as well as outliers or unexpected results.
5. Mitigation strategies: Based on the data analysis, it is possible to identify strategies for mitigating the impact of packet dropping attacks on data collection in IoT networks. These strategies may include changes to the network topology, improvements to the routing protocol, or the implementation of security measures.
6. Validation: Finally, the results of the analysis and mitigation strategies should be validated through additional simulations or real-world testing to ensure their effectiveness in preventing or mitigating packet dropping attacks.



During packet transmission, each node generates numerous PFEs according to the forwarding packet behaviors of its neighbors. We design a table named PFE Table (PFET) for each node to store the PFEs. PFET is shown in Table 2. where there are four fields: Packet-ID, Forwarding Node, Receiving Node, and Capacity. Packet-ID means the identifier of the forwarded packet; Forwarding Node means the node that forwards the packet; Receiving Node means the node that receives the packet; Capacity means the number of PFEs that a node can store. We assume that a node's total capacity is , and it is divided equally to its neighbours. According to Table 2, we can know that has generated three PFEs about , which, respectively, represent has forwarded to , to , and to .

Stage 1 Cooja Installation:

We will install the contiki file using the instant contiki section in the website and after downloading the file we will extract the file and run it in the VM box as cooja runs on Ubuntu further will be explained in the System design section.

Stage 2 Running cooja:

Now we need to run the VM box and login into the Ubuntu account. Then we need to open the cooja simulator and create a new simulation in it and save the file.

Stage 3 Implementation in Cooja:

After creating a new simulation in Cooja we need to arrange the sink mode and then the sender modes in the arrangement section of the simulator.

Stage 4 Analysis:

Now we need to start the simulation by clicking on the run button and running it for about 10 minutes. Then using the collect data view we need to collect data and analyse the graphs and node paths received in attack and no attack implementation.

Stage 5 Algorithm construction:

After analysis of the data we need to write an algorithm to detect and solve the problem of the Packet drop attack

Chapter -2 LITERATURE SURVEY

2.1 Related Work

A literature survey on the impact of packet dropping attacks in data collection in IoT using COOJA shows that several studies have focused on the analysis of packet dropping attacks and their impact on network performance and data collection in IoT.

One such study is "Impact of Packet Dropping Attack on Data Collection in Wireless Sensor Networks" by R. Iqbal et al. This study used simulations in COOJA to analyze the impact of packet dropping attacks on data collection in WSNs. The results showed that packet dropping attacks can cause significant data loss and delay, and that the impact of these attacks is greater in larger networks.

Another study, "A Comprehensive Study on Packet Dropping Attacks in IoT-Based Wireless Sensor Networks" by A. Sharma et al., also used COOJA simulations to analyze the impact of packet dropping attacks in IoT-based WSNs. The study found that packet dropping attacks can lead to a decrease in packet delivery ratio and an increase in end-to-end delay, and that these effects are more pronounced in networks with a higher degree of connectivity.

Additionally, a study by S. Gautam and N. Kumar, "An Experimental Analysis of Packet Dropping Attacks in IoT-Based Wireless Sensor Networks," used COOJA simulations to analyze the impact of packet dropping attacks on IoT-based WSNs that use the RPL routing protocol. The study found that packet dropping attacks can significantly impact the performance of the RPL protocol, resulting in a decrease in packet delivery ratio and an increase in end-to-end delay.

IT networks are helpless against various security assaults and the RPL convention influences this weakness, which causes the RPL-based Parcel organization to become inclined to RPLspecific or convention explicit and SN-acquired assaults. The examination in this space is still in its underlying stages, despite the few investigations led in regards to go after identification and anticipation in such organizations. RPL-based

Part networks are defenseless against both convention explicit and SN-acquired goes after at the same time, which consumes the RPL-based network assets and compromises secrecy, honesty, and accessibility (CIA) security set of three prerequisites [11].

In this segment the connected existing writing is assessed, which is introduced by analysts for assault recognition in RPL and IoT. In [12], the creators have examined late correspondence and organization conventions material in the IoT climate, while in [10], the creators have given a top to bottom examination of RPL-related security assaults, RPL piece, parts, and control messages. They have likewise introduced the assault characterization scientific classification and an organized order of countermeasures introduced by different specialists.

In [13], the creators have proposed an AI based double grouping technique to distinguish one of the convention explicit assault types. They have produced their dataset because of the absence of a fitting dataset by making a variant number assault network model to reenact the assault in the Cooja network test system and accumulate the information. In machine learning, highlight scaling and choice are two of the main advances and there are various strategies to play out these means referenced in the writing.

The creators have utilized a min-max scaling method and forward highlight determination strategy to preprocess their dataset. Moreover, a light slope supporting machine is utilized as a parallel classifier to identify the variant assault in a RPL network with an alternate number of hubs. The outcomes show that the proposed model performs particularly well in the classification of typical traffic from assault traffic. In any case, there is a hole distinguished as far as tending to SN-acquired assaults and thought of hub versatility.

The portability metric is referenced yet not examined. Essentially, in [14], the creators have proposed a profound learning-based model for the location of flood assaults. These assaults fall under the class of SN-acquired assaults. They have assessed the model utilizing precision and relapse related assessment measurements including mean squared mistake, mean outright blunder, and root mean squared blunder. The model is contrasted and different classifiers including support vector machines (SVM) and it performs well in examination.

Be that as it may, the convention explicit assaults are not viewed as in this examination and the versatility of Parcel hubs is likewise not tended to. Besides, profound learning techniques are notable for calling for greater investment and information, and their trouble in translation. In , the creators have proposed to utilize oneself getting sorted out map-based profound learning technique for fostering an interruption recognition framework to address RPL assaults.

In any case, the situation technique - which is a significant consider such arrangements - was not expressed plainly in this study [13]. In [16], the creators proposed to address RPL assaults including the position assault utilizing a profound brain network approach. They produced the dataset involving the Cooja test system in the Contiki working framework and considered the model in contrast to standard execution boundaries including precision, accomplishing the most elevated exactness for one assault called the welcome flood assault. In any case, this approach prompts comparable issues, as talked about prior in this part, i.e., long preparation time, and weakness to different assaults in the model layers. In [17], the creators have proposed a trust-based system to address security issues in RPL networks with an emphasis on convention explicit position assaults. They played out a reenactment study to assess the proposed expected transmission include metric-based methodology as far as energy utilization, parcel conveyance rate, throughput, and rank change in the organization.

Be that as it may, portability was not tended to. Furthermore, an equipment security chip is expected alongside the hubs. In [18], the creators have tended to transmission assaults in vehicular promotion ho Parcel networks utilizing a trust-based method. Be that as it may, the steering assaults were not thought of and it was restricted to vehicular impromptu organizations. Likewise, in [19], the creators have tended to security issues in vehicular promotion ho Parcel networks utilizing a trust-based convention for sticking assaults and recognizing vindictive hubs in such Part empowered networks. Nonetheless, RPL assaults were not viewed as in the proposed approach. Various audit studies and reviews have been directed investigating various methods to address the security assaults in Part and RPL-based Parcel. For example. in [20]. the creators have played out an orderly writing survey of ML and DL techniques for assault recognition in RPL-based

Part. Essentially, in [21], the creators have directed a nitty gritty overview for the assessment of RPL assaults. They have additionally evaluated different recognition and alleviation strategies utilizing RPL control messages. Table 1 presents a synopsis of the connected works notwithstanding the dataset(s) utilized, philosophy, assaults considered, and limits of exploration holes.

To give the best security system numerous analysts are as yet chipping away at IoT and remote sensor regions. In this section, explain different such interruption location frameworks, which are proposed lately, are depicted exhaustively. Le [7] proposed IDS to recognize geography assaults like position assault and nearby fix assaults. FSM (Limited state machine) approach is utilized in the proposed framework. The screen hub notices the way of behaving of adjoining hubs with FSM; in the event that any thought hub disrupts the position guideline. Then it recognizes it be an interruption. In the event that there is no variety in rank rule, the screen hub really looks at the way of behaving of the thought hub with the other screen hubs and recognizes the aggressor.

For location of nearby fix assault, one limit esteem utilized, and when the hub esteem surpasses the edge, an alert is raised of neighborhood fix assault. Dvir [4] et al. proposed a technique intended for VERA ("Rendition number and rank validation"). The hubs are having competent to change the position worth of a "DIO message" equivalent to it disregards every one of the further hubs. Which hub is close up to the root or source and neighbors that the position values are indicated. Then, to execute a position assault that rank worth could be duplicated through another hub. In VERA" technique forestalls the going after hubs as of getting lower rank worth than its unique position esteem by executing a one-way hash chain strategy, "VERA" technique is utilized to ensure that stringently builds the position from the "DODAG" root to the adjoining hubs. The VERA technique effectively distinguishes the "version number assault": however it actually doesn't work for two distinct geography assaults. First is satirizing of rank; second one is a position replay assault, a noxious hub to give answer to its folks proclaim that the position esteem is one level closer to the root. "TRAIL" et.al [5] was projected to stick issue is the most incomplete of the message, rank validation in "VERA" technique. "TRAIL" strategy presents improvement to "VERA" for nearby fix, and in conclusion find and disengage misleading hubs, those hubs assault the RPL directing order. "TRAIL"

strategy likewise has one issue. in which a youngster/kid hub picks an assailant hub as its parent hub, considering that a high schooler hub can't choose where that parent hub is an assailant hub or not. Consequently, Bothersome et al. projected a protected parent goal to ensure that youngster hubs select a reasonable hub as of their parent. S. Raza, L. Wallgren [6] proposed

"Smooth" approach, IDS that handle various assaults, as well as sinkholes, particular sending. This framework characterizes with three models: interruption detection, mapping and last min firewall to steering attacks. P. Kasinathan overview [3]. the creator portrays "Ebbits" approach, a construction that utilizes a module to notice the traffic of the organization in coordinates to execute an examination and distinguish making trouble hubs or noxious hubs. "Ebbits" identify the Disavowal of administration assaults DoS) in 6LoWPAN organization. A large portion of the Part includes are reasonable to "Ebbits" idea. Notwithstanding, Smooth [1][6] and "Ebbits"

result shows high asset utilization, produce low organization execution and framework execution. A particular based IDS approach is proposed by Le et.al, to put the issue in the

"Smooth" strategy, as given that low Misleading positive rate (FPR and low asset utilization. Table-1 portrays an examination among the strategy and IDS methods that are utilized to distinguish rank assault.

In our work, we carried out numerous immediate and circuitous assaults in RPL/6LoWPAN steering convention. RPL is the principles steering convention for the web of things. With the gigantic associated numbers today, the requirement for security is sought after. The Parcel network is interfacing billions of gadgets and producing enormous heterogeneous information. In our work. we showed what these assaults work and demonstrated their mean for on execution measurements. We contrasted the impact with the standard reenactment. In this manner, it is our gauge reference, consequently playing out a full scope of examinations of gone after parts of organization (assets, geography, and related information streams) in different circumstances. We checked on the outcomes acquired to assist with characterizing which boundaries ought to be followed or referred to for the applied assaults. We will utilize the two arrangements of results and distinguished marks of give and take nitty gritty in this paper with respect to our future work. We work on a framework engineering where we will be founded on man-made reasoning to identify, forestall and decide appropriate countermeasures for these assaults RPL is presently applied in the job of a primary steering convention for enormous scope low-power and lossy organizations, and in this way turns into a decent possibility to bring the utilization of WSN into various regions like Trap of Things or Splendid Organization. It is basic to Get the association execution and will before long turn into a significant necessity to coordinate into such applications. The exceptional idea of "Rank" in RPL can cause its presentation to become defenseless against the inner dangers. Assault on Rank might make un-advanced ways, all the more above, and more parcel crashes, subsequently minimizing the organization execution by, for instance, expanding the start to finish delay and diminishing the conveyance proportion. In this

paper, we initially examine some conceivable Position assault dangers that can be executed to minimize RPL execution. We then, at that point, concentrate on the effect of the assault by recreating assaults on various areas inside the organization.

The outcomes uncover that assault might seriously affect the organization execution, particularly when it is carried out in a high sending load region, or in different aggressor cases. Various kinds of Rank assault conduct are dissected regarding concealing the non-streamlined steering data or flipping the favored guardians. Our concentrate likewise uncovers that the quantity of impacted hubs, number of DIO produced, start to finish postponement and conveyance proportion are the most delicate to this specific sort of assault. Moreover, the outcomes show a shortcoming in the security plan of RPL with the goal that the youngsters need to depend on its parent's directing data through DIO bundles however they have no other component to confirm the administrations of their folks. It is critical on the grounds that once the favored parent is compromised and no other hub finds its malevolent way of behaving, the presentation of all the encompassing region will be impacted. The discoveries likewise recommend that it is vital to present greater security assets in a few critical pieces of the organization than others in light of the fact that the assault has an alternate degree of effect in various organization regions. Later on, we might want to extend the aftereffects of this review for executing an inconsistency interruption identification framework, which can analyze the interior assaults in view of checking some assault delicate execution boundaries.

There are many advantages of integrating Part into various applications. Notwithstanding the advantages, it is powerless against different dangers. Thus, working on the security of the Parcel network is vital. The proposed E-RAD calculation consolidates a rate limit on age of DIO control parcels and the sales of DIS control bundles for ID of recently joined assailants. In the event that aggressors are missed in the above situation, it is identified from consistency check of hash esteem. Upon the ID of rank aggressors, a caution is created to seclude them from DODAG. Caution is a connection in the DIO message itself. Cooja Test system is utilized to survey the power of the E-RAD calculation. The calculation accomplishes 95.53% of PDR, 97.23% of exactness, normal

postponement of 775msec, 1023 number of control parcels in matrix focused. The assessed energy utilization is brought down to 1356 joules. In the vast majority of the current frameworks and the assessment of E-RAD static geography is thought of. Support for portability is critical in numerous IoT-based applications. In this way, the E-RAD calculation can be extended to consolidate hub portability and ID of other Part aggressors.

Chapter-3 SYSTEM DEVELOPMENT

3.1 Background

The development of systems for analyzing the impact of packet dropping attacks in IoT builds on the growing use of IoT technologies in various applications. With the increasing adoption of IoT devices in homes, factories, and other settings, it has become critical to ensure the security and reliability of these systems.

The development of this system involves the use of COOJA, a network simulation tool, to model and analyze the impact of packet dropping attacks in IoT networks that use the RPL routing protocol. This system aims to provide insights into the performance of IoT networks under different attack scenarios and configurations, as well as identify strategies for preventing or mitigating the impact of packet dropping attacks on data collection.

To develop this system, researchers must have a background in IoT technologies, network protocols, and security. Additionally, expertise in using simulation tools like COOJA is necessary to create an accurate model of the network and simulate various attack scenarios. Knowledge of programming languages such as C and Java is also essential to develop custom scripts and applications to run on the simulated network.

In a Packet Dropping attack, a malicious node inserts a bogus Packet through DIO onto the resulting nearby node that has converged towards it and has been chosen as a preferred parent through DAO. Once the malicious node becomes the attacking region's preferred parent node (PP), network performance may gufler (7). Afler the topology has been established, the attacker introduces one rogue node in order to begin a packet dropping attack on the BFL network.

All nodes, as determined by the defined OF obey rank rules, choose the PPN with the lowest Packet. Different paths in the network are constructed in accordance with typical criteria.

We will be implementing everything on Cooja software. The figure 3.1a explains the working of the cooja simulator.

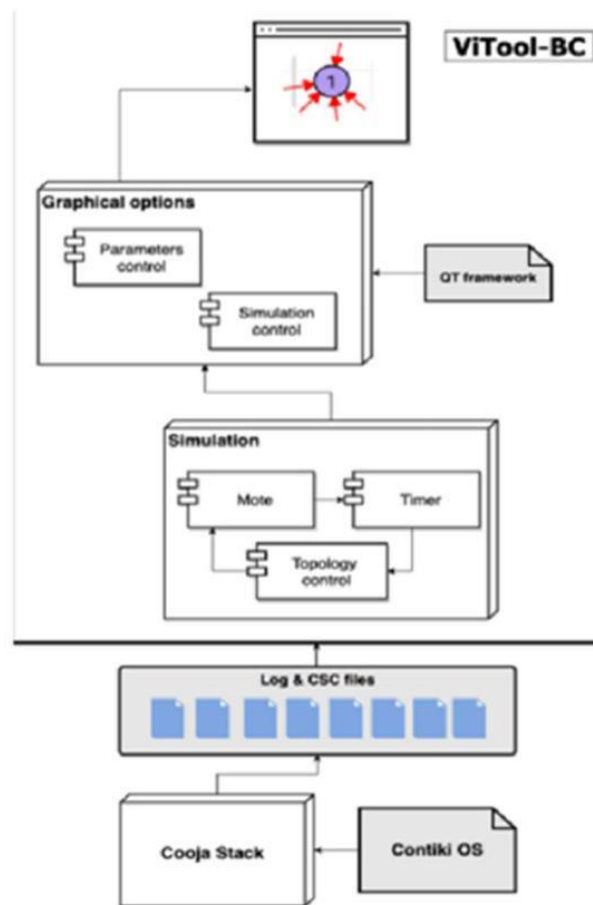


Figure 3.1a Cooja Simulator flow diagram

What is Contiki?

An IoT operating system called Contiki is designed in particular for tiny IoT devices with constrained memory, power, bandwidth, and computational capacity. It has a simple design but nevertheless has all of the typical tools seen in contemporary operating systems. It offers features for managing communications, resources, processes, programmes, and memory.

Standard protocols and more current enabling protocols for IoT are supported by Contiki.

- (For IPv4) uIP Both 8-bit and 16-bit microcontrollers are supported by this TCP/IP implementation.
- This is a completely conforming IPv6 extension to uIP called uIPv6 (for [Pv6]).
- When IPv4 or IPv6 are impractical, Rime offers an alternative stack as a workaround. It provides a collection of low-power system primitives.
- IPv6 over low-power wireless personal area networks is referred to as 6LoWPAN. It offers compression technologies to enable the slow wireless data rates required by resource-constrained devices.
- RPL enables the optimum path to be identified in a complicated network of devices with a variety of capabilities in LLNs (low-power and lossy networks).
- CoP - This protocol enables communication for straightforward devices, typically those that require intensive remote control.

Contiki

The Open Source OS for the Internet of Things

Figure 3.1b Contiki OS

What is Cooja?

COOJA is a network simulator that is widely used in the development and testing of wireless sensor networks (WSNs) and internet of things (IoT) systems. It is an open-source simulator that is part of the Contiki operating system, which is a lightweight operating system designed for IoT devices.

COOJA allows developers to create a virtual simulation of a wireless network and test various network protocols and applications in a controlled environment. It supports several network protocols, including the popular RPL (Routing Protocol for Low-Power and Lossy Networks), and provides various debugging and visualization tools to analyze network performance.

In COOJA, developers can create a simulation of a network using a graphical interface and add various nodes, each representing a sensor or IoT device. They can then define the behavior of each node, including its energy consumption, communication range, and packet routing behavior, among other parameters.

Using COOJA, developers can simulate various scenarios, such as network congestion, node failures, and security attacks, to test the performance of the network and identify potential issues or vulnerabilities. This allows them to optimize the network protocols and applications before deploying them in a real-world environment.

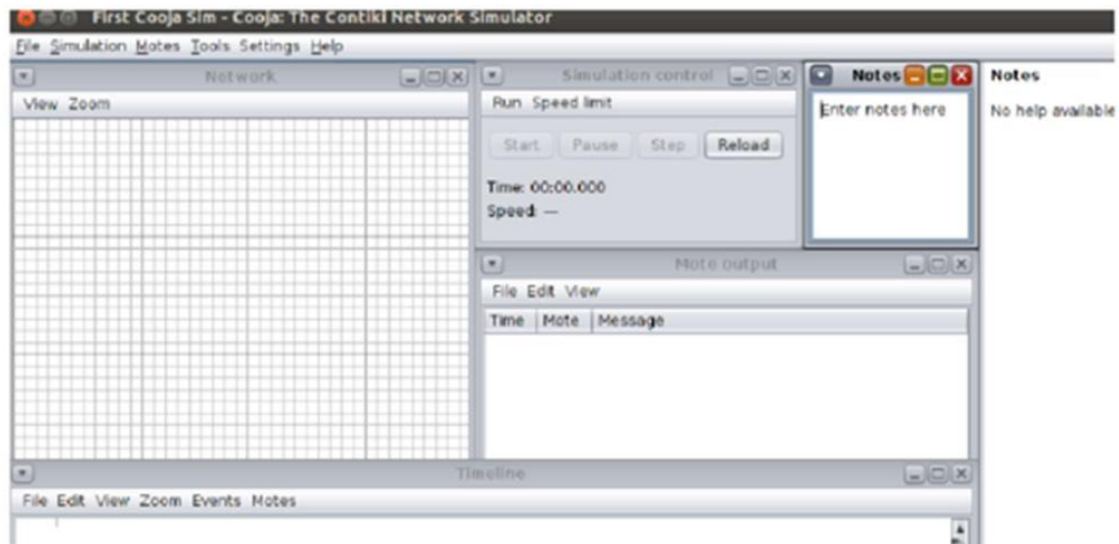


Figure 3.1b Cooja Simulator UI.

What is Virtual Machine and why do we need it to run cooja?

On a physical hardware system, a virtual machine (VM) is a virtual environment that performs as a virtual computer system with its own CPU, memory, network interface, and storage (located off- or on-premises). The resources of the system are separated from the hardware and provisioned properly so that the virtual machine (VM) can use them. This software is known as a hypervisor.

Because Cooja is a simulator for the Ubuntu operating system, we need to use a virtual machine (VM) as a second host in order to run Cooja.

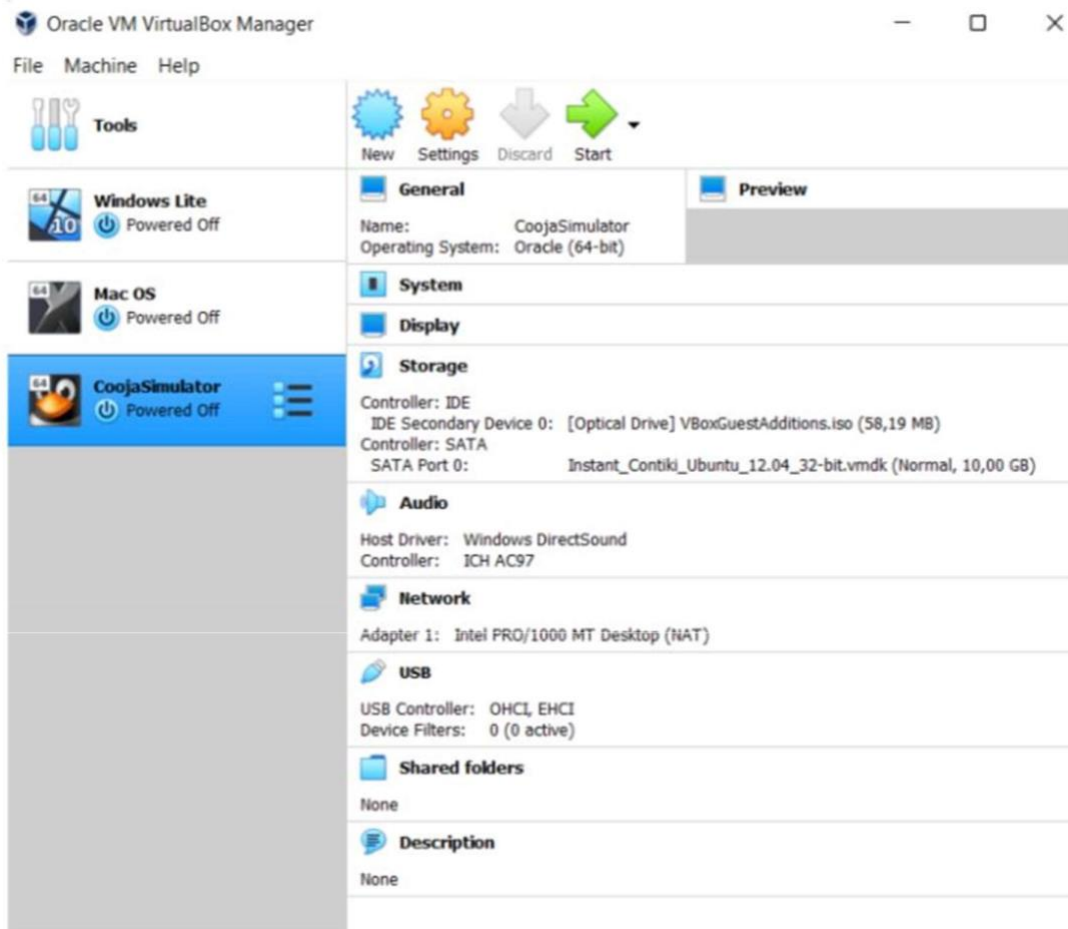


Figure 3.1c Virtual Machine with cooja Simulator.

3.2 Implementation

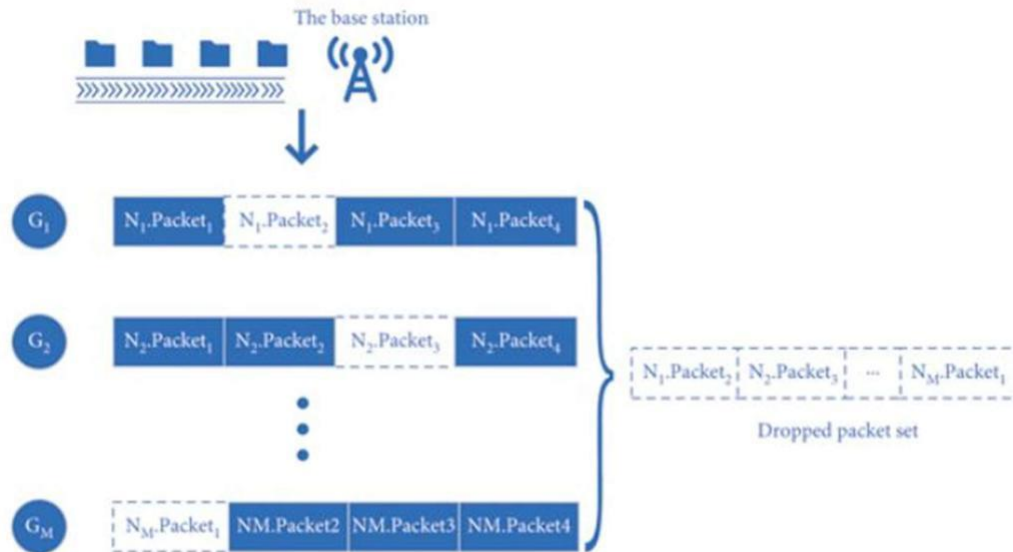


Figure shows, the base station divides the received packets into groups and sorts the packets for each group. For the first group of the source node, the base station receives the packets with sequence number, , and except . So it finds that is dropped. After checking all groups, the base station can obtain the dropped packet set.

3.2.1 Installing Cooja on your Operating System:

Step1: We need to first install an Virtual Machine on our machine so that we can run

Ubuntu as an operating system on the host machine and run the cooja simulator on in.

Step3: After extraction we need to create a new virtual machine named cooja and choose the appropriate file from the extracted folder as the disk file. In the figure 3.2.1c we can see the highlighted file that needs to be selected as the disk file while creating the virtual machine for cooja in the VM.

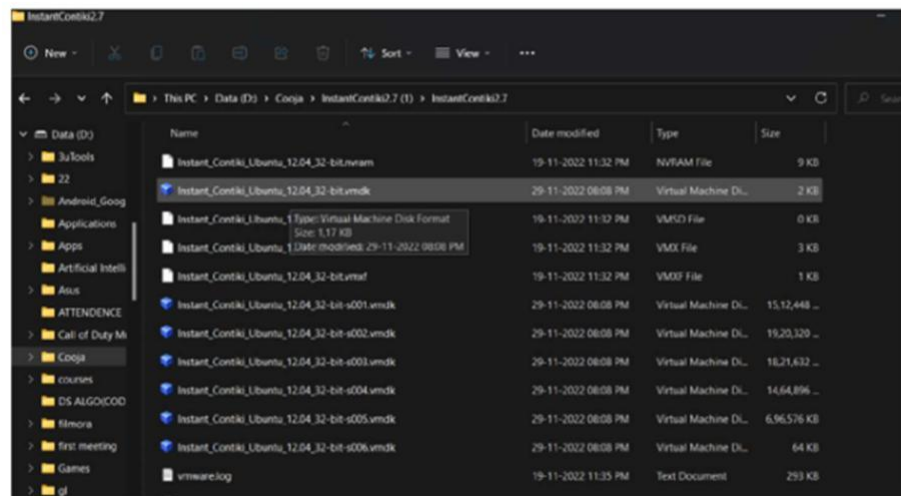


Figure 3.2.1c Cooja disk file.

Step4:Run the virtual machine by choosing the cooja machine and clicking on the start button.

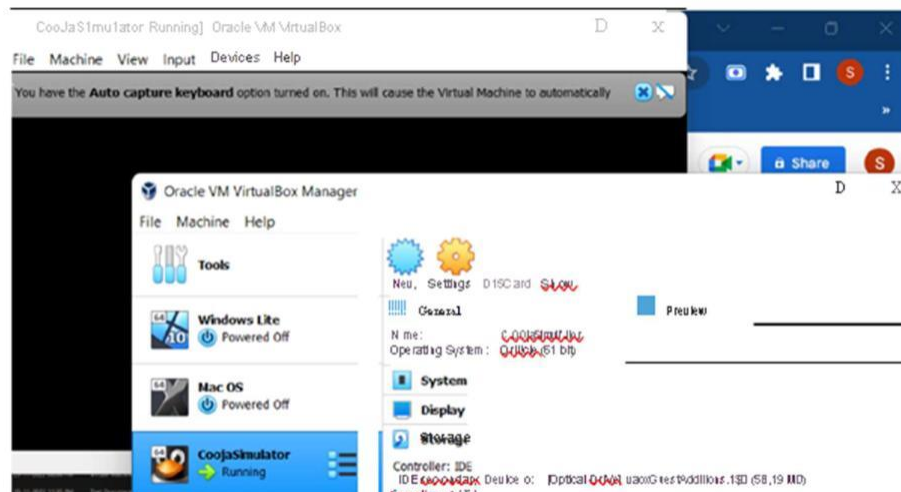


Figure 3.2.1d Starting the virtual machine of Cooja.

Figure 3.2.Id Starting the virtual machine of Cooja

Steps5: It will ask for an password to login. The default password for all cotinki cooja simulator is user.

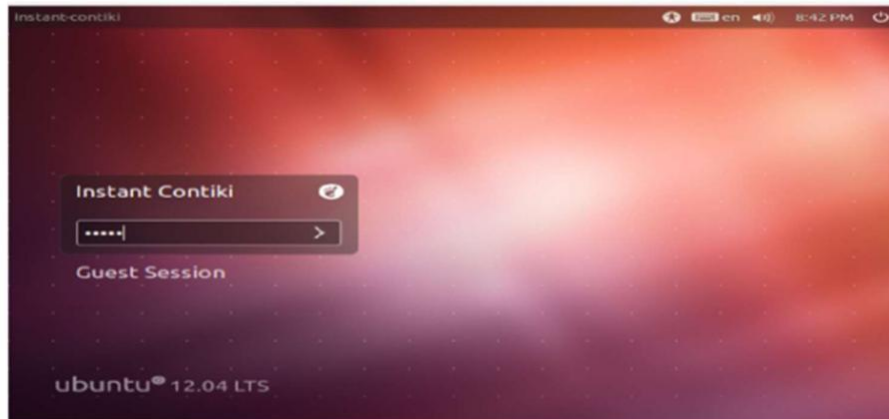


Figure 3.2.1e Password field in VM of Coojauser

Figure .1f shows the VM screen we get after proper installation of CotinkiOS.



Figure 3.2.1f Cotinki screen

Fig. Implementing the no attack data collection structure in Cooja

Step1 :After opening the ubuntu OS on VM and logging in we now need to open the cooja simulator by clicking on it.

Step2: We will see something like in the figure 3.2.2a.So click on the file options and choose new simulation option from the menu as shown in the figure.

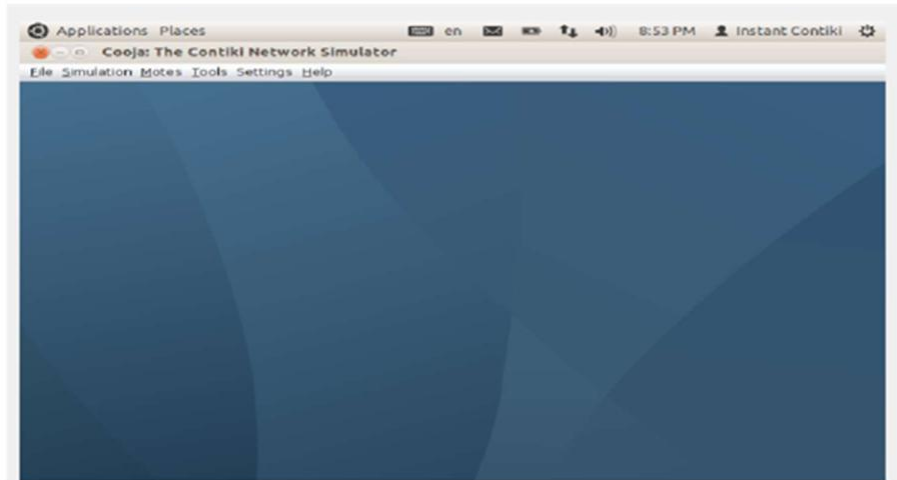


Figure 3.2.2a Cooja initial screen

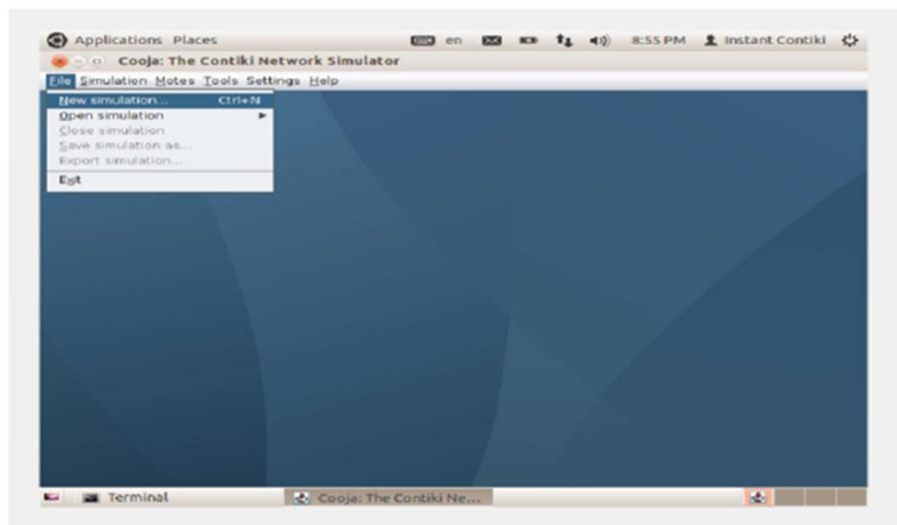


Figure 3.2.2b Creating new stimulation.

Step3: Making sure we on all the setting as shown in the figure 3.2.2c.

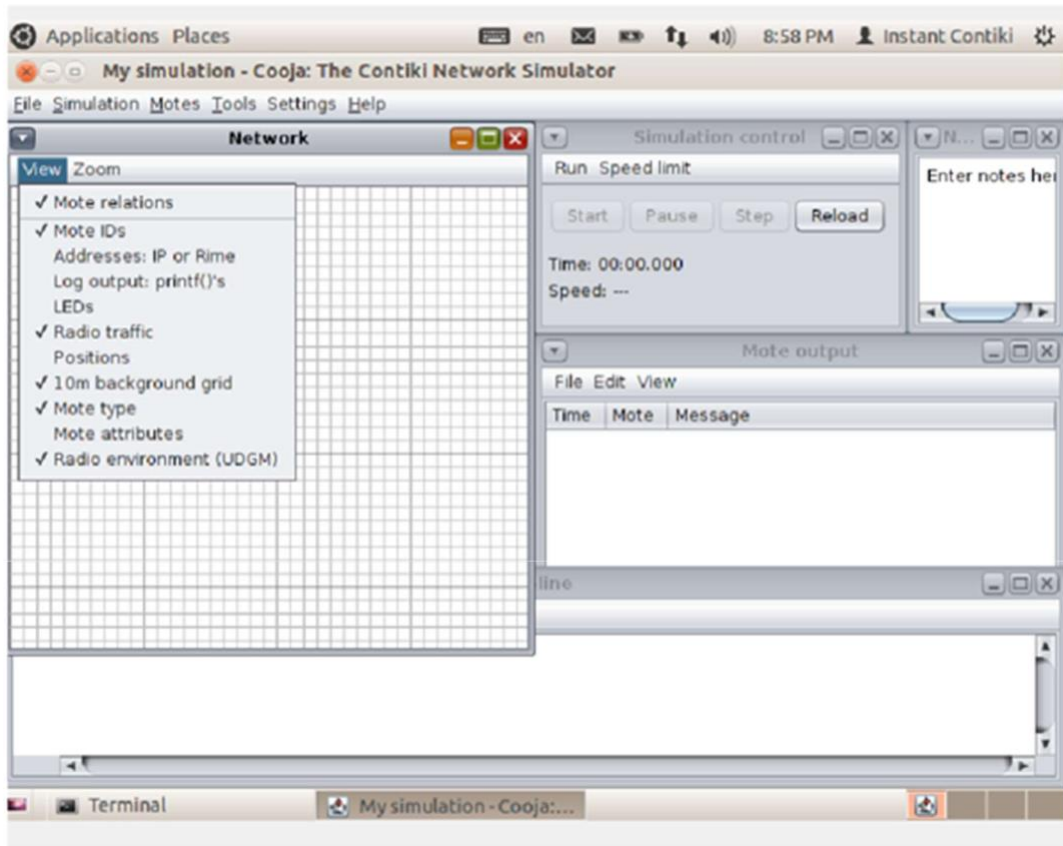


Figure 3.2.2c Important cooja settings

Step3: Now we need to create the sink mode using the modes option and click on the modes dropdown and select add modes. Then we need to add sky mode and then browse the updsink.c file in cotinki->example->ipv6->rpl-collect->udpsink.c.

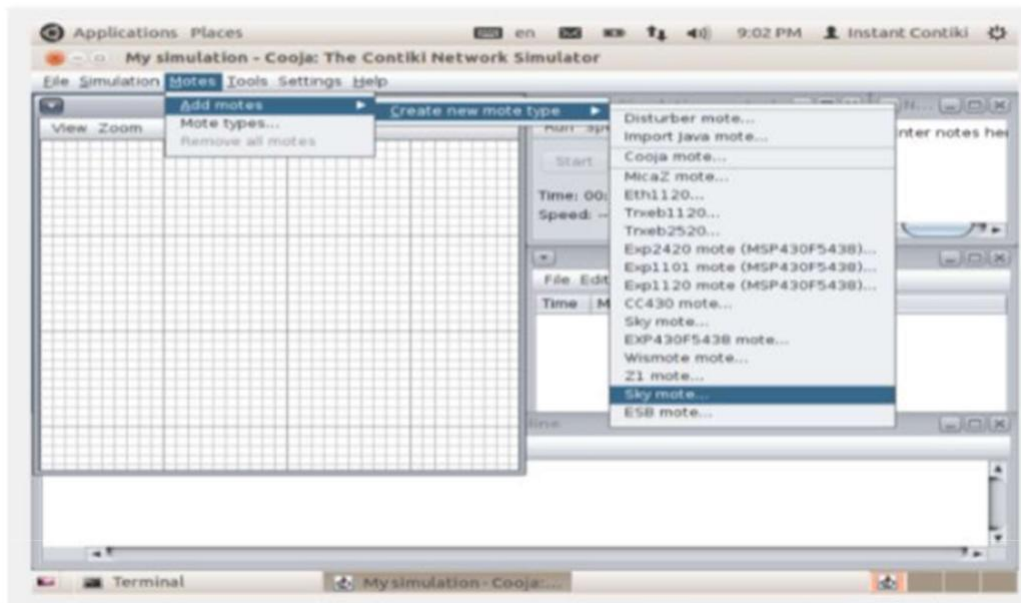


Figure 3.2.2d Adding skymode.

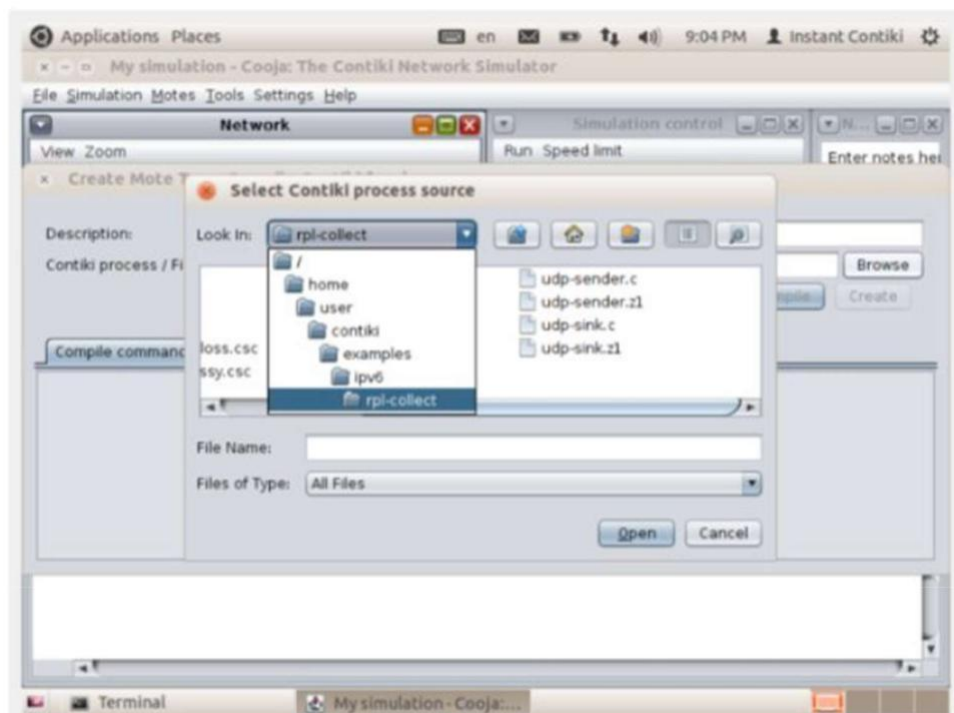


Figure 3.2.2e Selecting the udp-sink.c file.

Step4: Clicking on the compile option to compile the udp-sink.c code and then clicking on the create option and adding the mote we should we something like in the figure 3.2.2f.

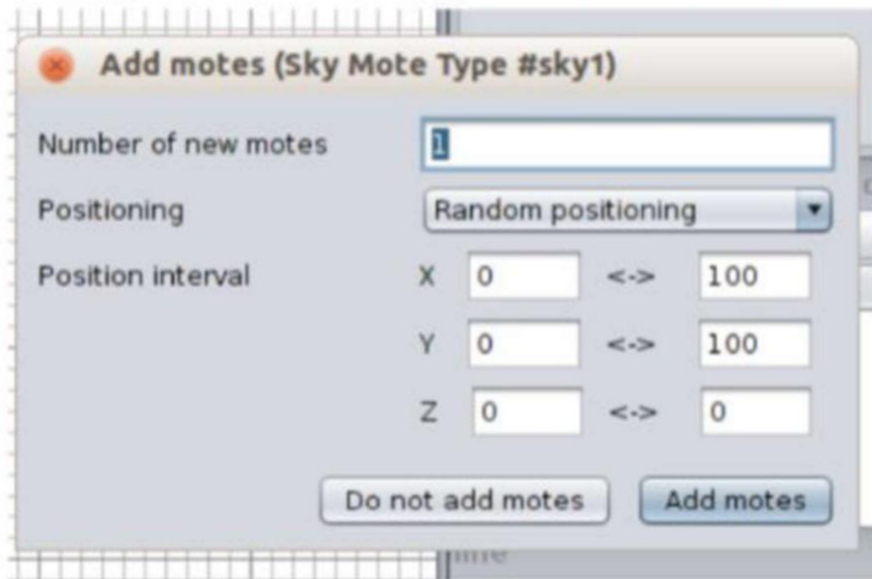


Figure 3.2.2f Adding the created mode node

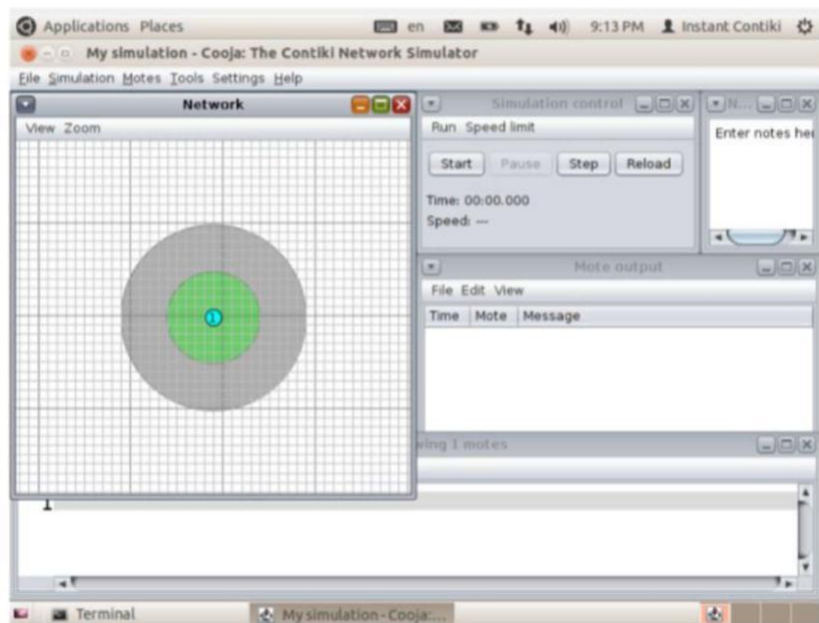


Figure 3.2.2f Added the created mode node.

Step5: Now we need to add other sender and receiver modes by following the step3 but by choosing the upd-sender. file. Then we need to repeat the step4.

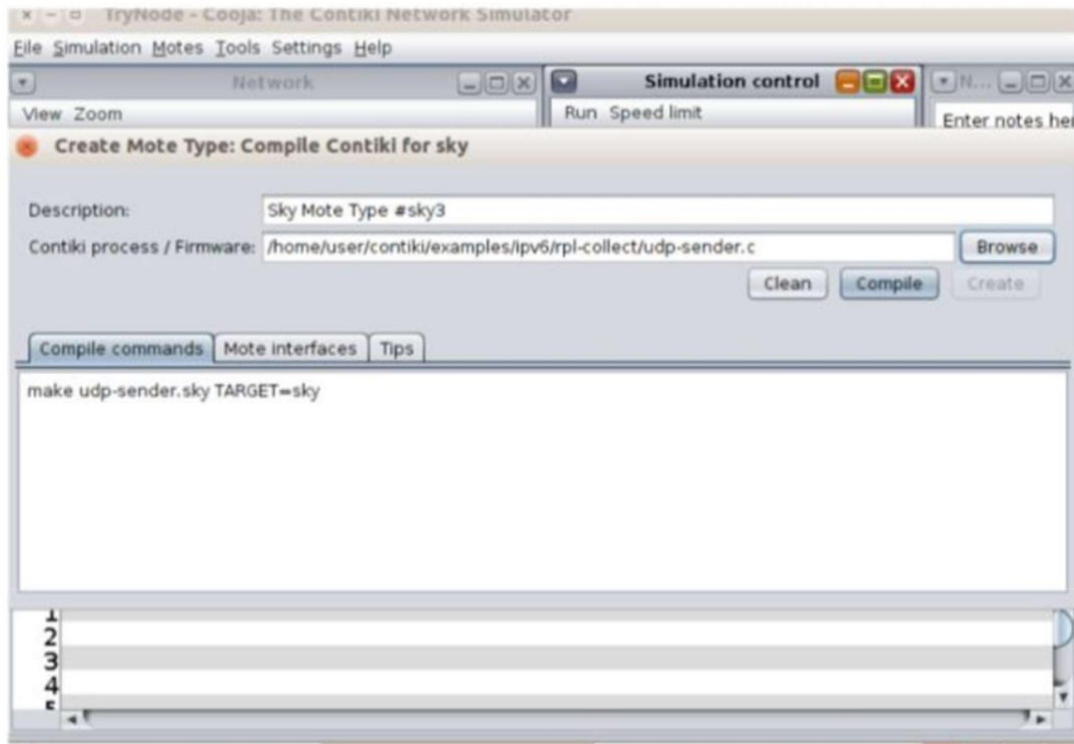


Figure 3.2.2g Adding the created mode node.

Step6: Now we need to arrange our added motes according to the structure we need. We will be arranging them in tree form as we need to implement rank wise data collection. So it will look like as shown in the figure 3.2.2h.

Step7: Now we need to click on the mode number I (sink mode) and from the options we need to select collect view so that we can collect data and analysis for the no attack rank structure implementation.

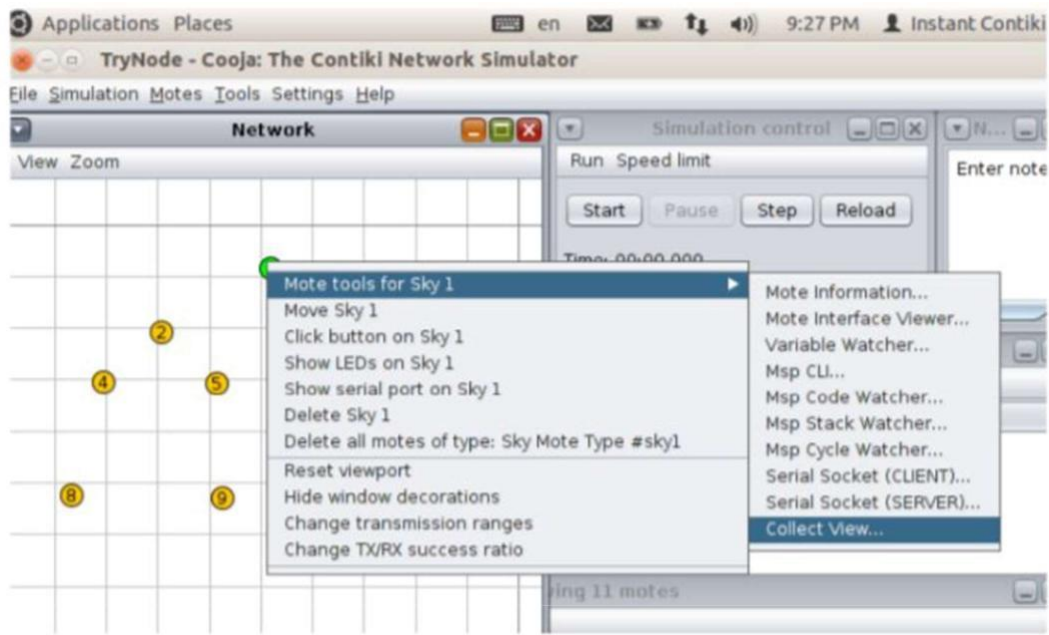


Figure 3.2.2h No attack rank structure.

Step8: Now in the collect view screen as in figure 3.2.2j we need to click on the Start Collect option and when the loading finishes then on the Send command to nodes option. Then we need to minimise the screen of Collect view and click on the start option of the Simulation control to start the simulation.

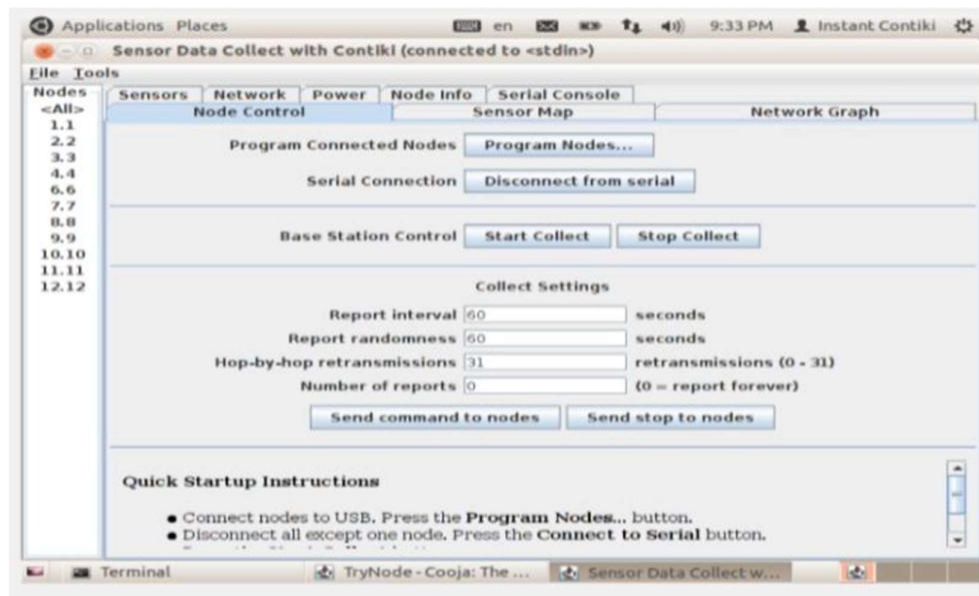


Figure 3.2.2j Collect View screen

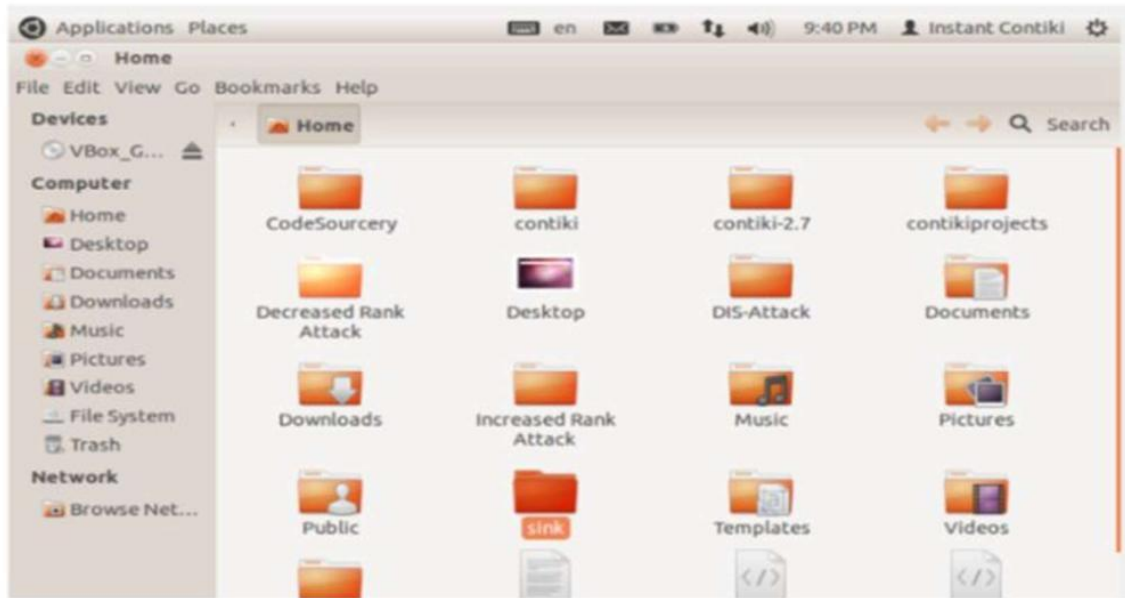


Figure 3.2.2k Starting simulation by start button on Simulation Control panel.

We have started the simulation and it will be collecting data for the no attack based structure and we will analyse the data with packet attacked stimulation

3.2.3 Implementing the Packet dropping attack data collection structure in Cooja:

Step1: We need copy the cotinki folder and rename it to decreased Packet drop attack and make changes in some files code to implement it.

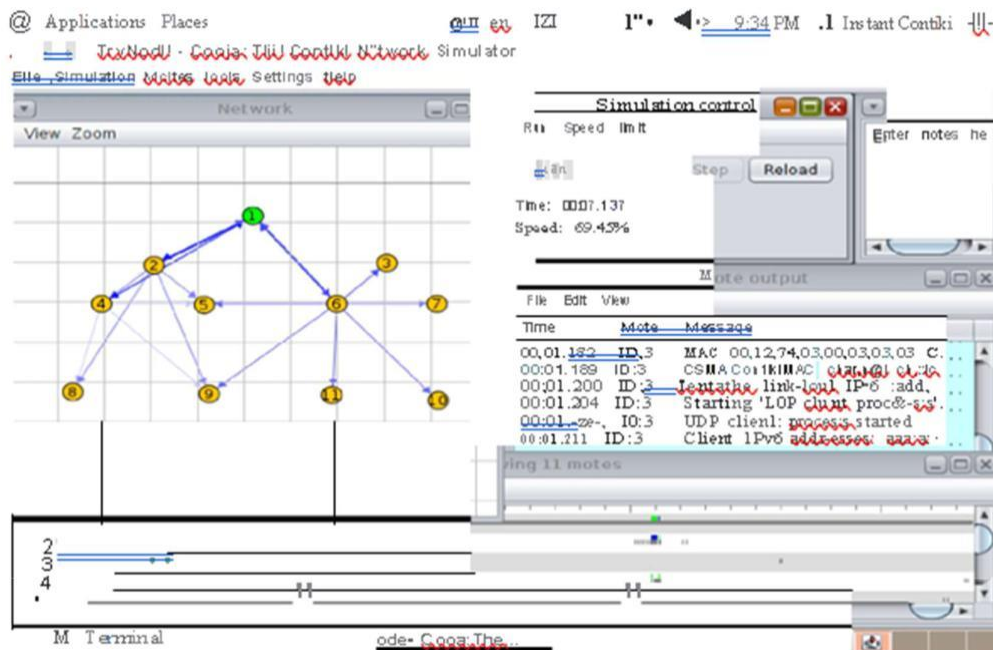


Figure 3.2.3a Decreased Packet dropping Attack folder created.

Step2: We need to go to the decreased Packet drop attack folder and then in core->net->rpl->rpl-private.h make the changes in the code as shown in the figure


```

rpl-private.h (~./Decreased Rank Attack/core/net/rpl) - gedit
File Edit View Search Tools Documents Help
Dev Open Save Undo
Com rpl-private.h
/*ifndef RPL_CONF_MIN_HOPRANKINC
#define RPL_CONF_MIN_HOPRANKINC 0 //changed
#define RPL_MIN_HOPRANKINC 256
#else
#define RPL_MIN_HOPRANKINC RPL_CONF_MIN_HOPRANKINC
#endif
#define RPL_MAX_RANKINC 0 //changed (7 *
RPL_MIN_HOPRANKINC)
#define DAG_RANK(fixpt_rank, instance) \
((fixpt_rank) / (instance)->min_hoprankinc)
/* Rank of a virtual root node that coordinates DAG root nodes. */
#define BASE_RANK 0
/* Rank of a root node. */
#define ROOT_RANK(instance) (instance)->min_hoprankinc
#define INFINITE_RANK 256 //changed 0xffff
/* Represents 32 bits */
C/C++/ObjC Header b Width: 8 Ln 118, Col 32 INS
"rpl-private.h" selected (12.1 kB)
[Terminal] [TryNode - Coo]... rpl rpl-private.h (~./...

```

Figure 3.2.3b Changes in the code of rpl-private.h

Step3: We need to go to the decreased Packet drop attack folder and then in core->net->rpl->rpl-timers make the changes in the code as shown in the figure 3.2.3c.

```

rpl-timers.c (~./Decreased Rank Attack/core/net/rpl) - gedit
File Edit View Search Tools Documents Help
Dev Open Save Undo
Com rpl-timers.c
/* dlo_send_ok is true if the node is ready to send DIOs */
static uint8_t dlo_send_ok;
/*
-----
*/
static void
handle_periodic_timer(void *ptr)
{
    rpl_purge_routes();
    //rpl_recalculate_ranks();
    /* handle DIS */
    #ifdef RPL_DIS_SEND
    next_dis++;
    if(rpl_get_any_dag() == NULL && next_dis >= RPL_DIS_INTERVAL) {
        next_dis = 0;
        dis_output(NULL);
    }
    #endif
    ctimer_reset(&periodic_timer);
}
C Tab Width: 8 Ln 69, Col 5 INS
"rpl-timers.c" selected (8.4 kB)
[Terminal] [TryNode - Coo]... rpl rpl-timers.c (~./...

```

Figure 3.2.3c Changes in the code of rpl-timers.c

The following changes are made so that we can implement the decreased Packet drop attack. The `rpl-recalculate-rank` is disabled so that rank cannot be recalculated again in `rpl-timer.c` and other changes are made so that we can get an decreased rank of the mode we create than it is expected.

Step4: We need to open the simulation created in the no attack implementation and need to add a new mode that is the affected mode of decreased Packet drop attack by deleting an mode of the structure lets say mode 6 and add the malicious mode over there. We need to go to decreased Packet drop attack >example->ipv6->rpl-collect->udpsender.c. and compile the code and

create the defected mode and replace it in place of mode 6.

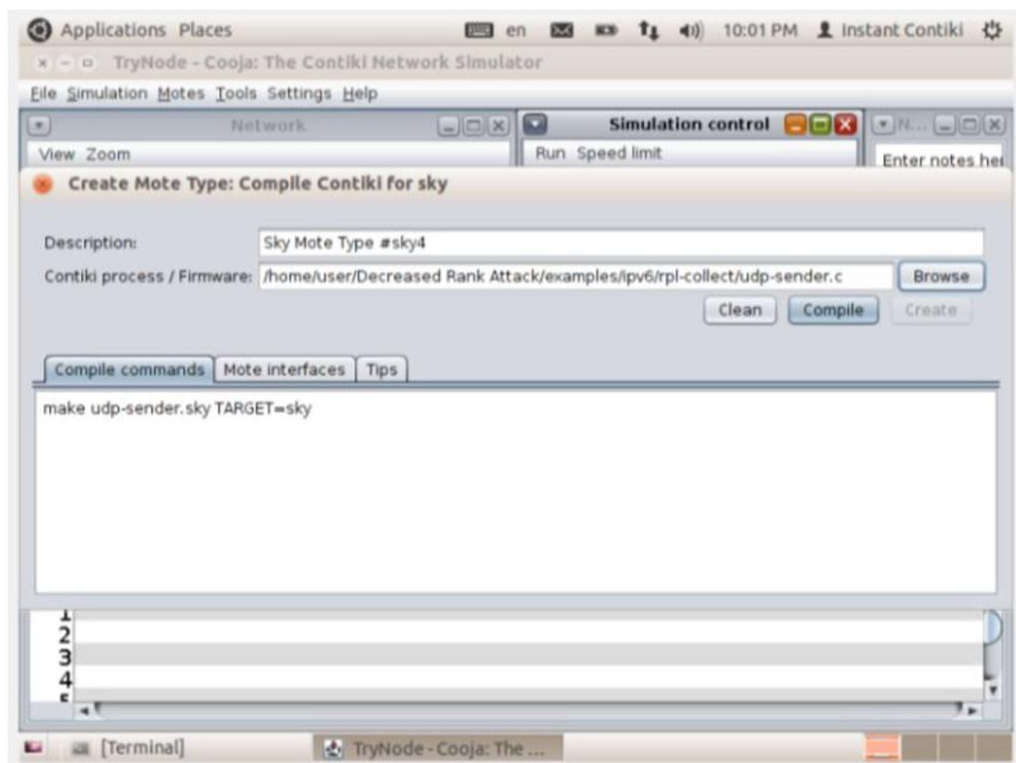


Figure 3.2.3d Decreased Packet drop attack data collection structure

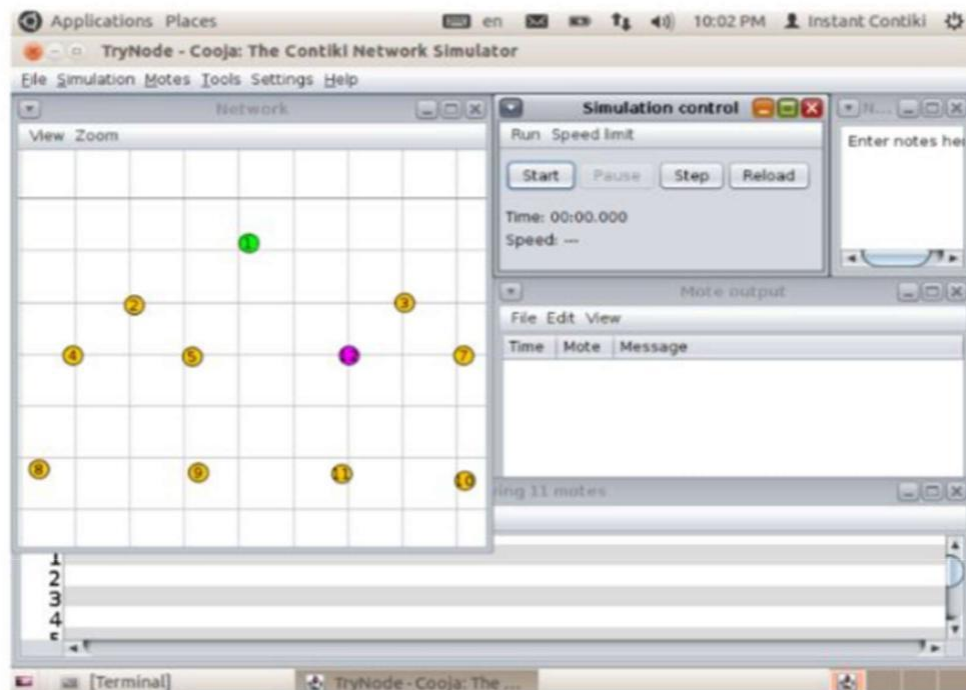


Fig 3.2.2 ie Implementing the no attack data collection structure in Cooja.

Step5: Now we need to click on the reload button on the simulation control panel and then repeat the steps 6 and 7 of the section 3.2.2 ie Implementing the no attack data collection structure in Cooja. We will now have the data and analysis for the decreased Packet drop attack data collection system.

3.2.4 Implementing the increased Packet dropping attack data collection structure in

Cooja:

Step1: We need copy the cotinki folder and rename it to increased Packet drop attack and make

changes in some files code to implement it as in figure 3.2.4.

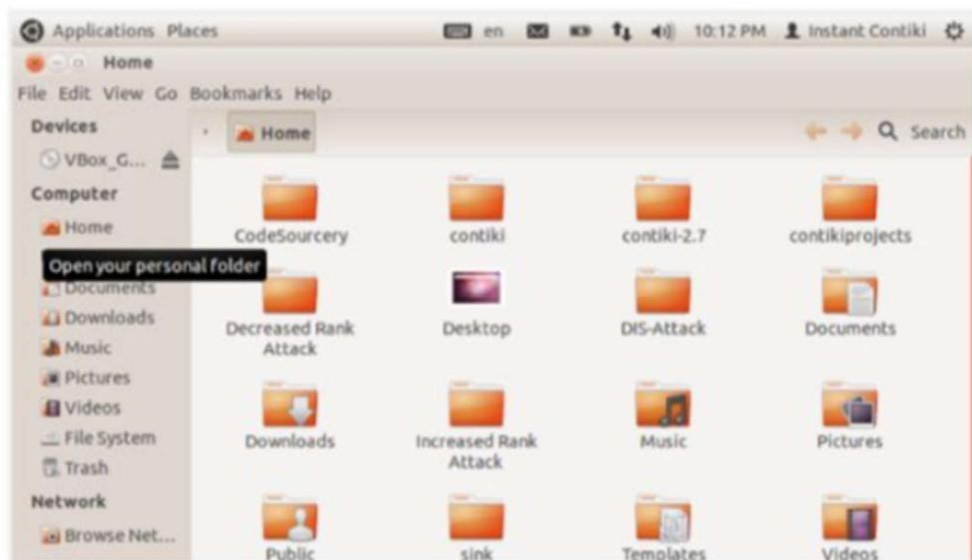


Figure 3.2.4a Increased Packet drop attack folder created.

Step2: We need to go to the increased Packet drop attack folder, and then in core->net->rpl->rpl-of0.c make the changes in the code as shown in the figure 3.2.4b.

```

rpl-of0.c
increment = p != NULL ?
    p->dag->instance->min_hoprankinc :
    DEFAULT_RANK_INCREMENT;

if((rpl_rank_t)(base_rank + increment) < base_rank) {
    PRINTF("RPL: OF0 rank %d incremented to infinite rank due to
wrapping\n",
    base_rank);
    return INFINITE_RANK;
}
return base_rank + increment+1; //change made

static rpl_dag_t *
best_dag(rpl_dag_t *d1, rpl_dag_t *d2)
{
    if(d1->grounded) {
        if (!d2->grounded) {
            return d1;
        }
    } else if(d2->grounded) {

```

figure 3.2.4b.

Step3: We need to go to the increased Packet drop attack folder and then in core->net->rpl->rpl-icmp6. make the changes in the code as shown in the figure 3.2.4c.

```

rpl-icmp6.c
pos = 0;

buffer = UIP_ICMP_PAYLOAD;
buffer[pos++] = instance->instance_id;
buffer[pos++] = dag->version;

#ifdef RPL_LEAF_ONLY
    PRINTF("RPL: LEAF ONLY DIO rank set to INFINITE_RANK\n");
    set16(buffer, pos, INFINITE_RANK);
#else /* RPL_LEAF_ONLY */
    set16(buffer, pos, dag->rank++); // change made
    PRINTF("Change made\n");
#endif /* RPL_LEAF_ONLY */
pos += 2;

buffer[pos] = 0;
if(dag->grounded) {
    buffer[pos] |= RPL_DIO_GROUNDED;
}

buffer[pos] |= instance->mop << RPL_DIO_MOP_SHIFT;
buffer[pos] |= dag->preference & RPL_DIO_PREFERENCE_MASK;

```

Figure 3.2.4c Changes in the code of rpl-icmp6.c

The following changes are made in code by increasing the rank of the mode so that the increased Packet drop attack can be implemented.

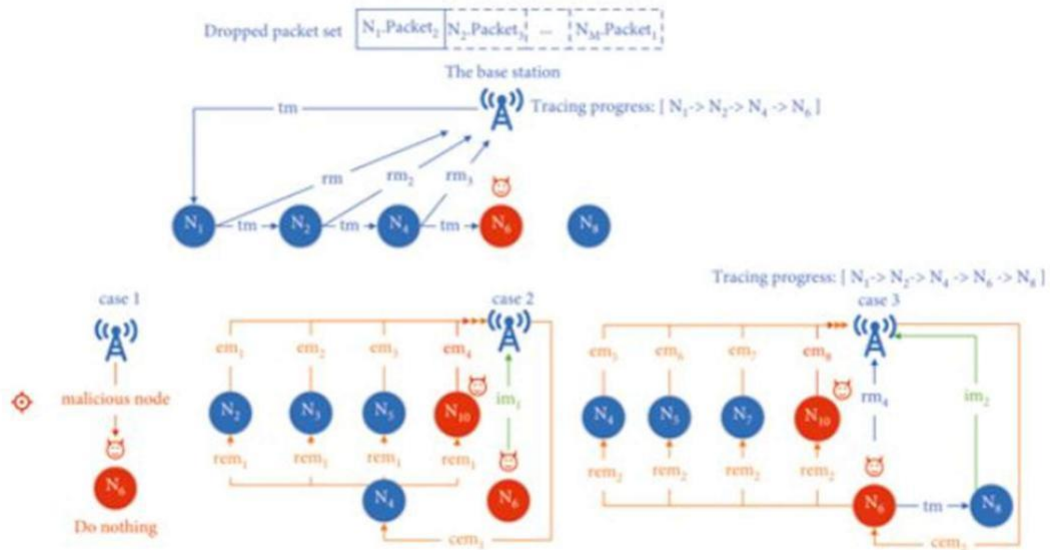
Step4: We need to open the simulation created in the no attack implementation and need to add a new mode that is the affected mode of decreased Packet drop attack by deleting an mode of the structure lets say mode 6 and add the malicious mode over there. We need to go to increased Packet drop attack >example->ipv6->rpl-collect->udpsender.c. and compile the code and create the defected mode and replace it in place of mode 6.

Step5: Now we need to repeat the step 5 of the section 3.2.3 Implementing the decreased Packet drop attack data collection structure in Cooja. We will now have the data and analysis for the increased Packet drop attack data collection system.

Chapter-4 Performance Analysis

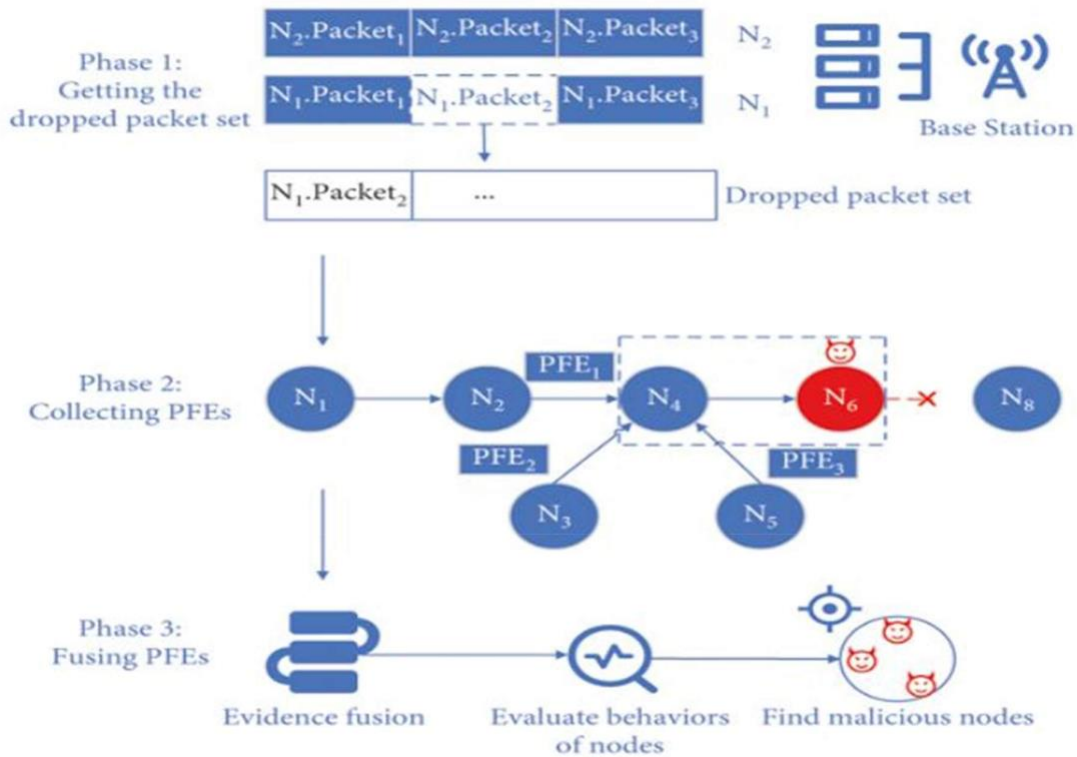
4.1 Comparing and Analysing the Data of the two stimulations

In this section we will analyse the data collected from all the two simulations no attack rank and decreased Packet drop attack structure. We will be comparing the graphs and paths followed by nodes and packets in the no attack and attacked stimulation.



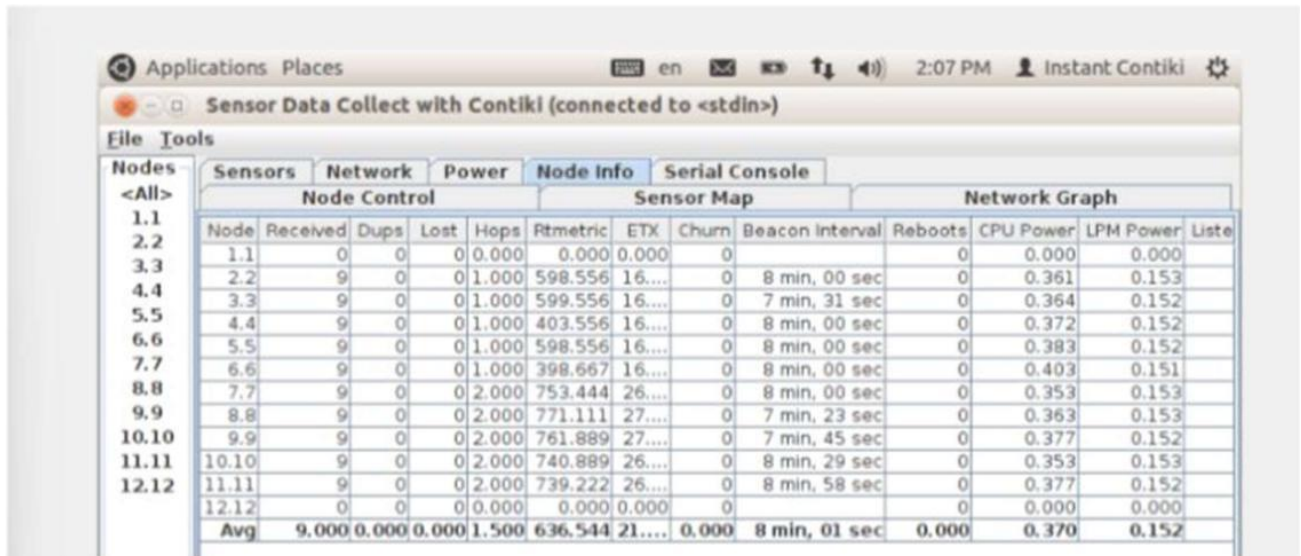
the base station finds the dropped packet and its corresponding source node . Then, it sends a message to and initializes the tracing progress as . Once receiving , node searches its PFRT and finds the next forwarding node is . It sends a message to the base station and forwards to

. When the base station receives , it updates the tracing process as . Once receiving , node continues to trace the routing path of the packet. After several steps of tracing, the tracing progress is updated to , and receives from . We assume that node is a malicious node



As mentioned above, EFDA does not need to inject extra packets to obtain the training dataset to train the detection model, and it utilizes the existing PFEs in the network to perform logical reasoning and identify malicious nodes.

4.1.1 Comparing the node information and sensor maps



Sensor Data Collect with Contiki (connected to <stdin>)												
File Tools												
Node Control				Sensor Map				Network Graph				
Node	Received	Dups	Lost	Hops	Rtmtrc	ETX	Churn	Beacon Interval	Reboots	CPU Power	LPM Power	Liste
1.1	0	0	0	0.000	0.000	0.000	0	8 min, 00 sec	0	0.000	0.000	
2.2	9	0	0	1.000	598.556	16....	0	8 min, 00 sec	0	0.361	0.153	
3.3	9	0	0	1.000	599.556	16....	0	7 min, 31 sec	0	0.364	0.152	
4.4	9	0	0	1.000	403.556	16....	0	8 min, 00 sec	0	0.372	0.152	
5.5	9	0	0	1.000	598.556	16....	0	8 min, 00 sec	0	0.383	0.152	
6.6	9	0	0	1.000	398.667	16....	0	8 min, 00 sec	0	0.403	0.151	
7.7	9	0	0	2.000	753.444	26....	0	8 min, 00 sec	0	0.353	0.153	
8.8	9	0	0	2.000	771.111	27....	0	7 min, 23 sec	0	0.363	0.153	
9.9	9	0	0	2.000	761.889	27....	0	7 min, 45 sec	0	0.377	0.152	
10.10	9	0	0	2.000	740.889	26....	0	8 min, 29 sec	0	0.353	0.153	
11.11	9	0	0	2.000	739.222	26....	0	8 min, 58 sec	0	0.377	0.152	
12.12	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	
Avg	9.000	0.000	0.000	1.500	636.544	21....	0.000	8 min, 01 sec	0.000	0.370	0.152	

Figure 4.1.1a Node information of no attack stimulation.

Node	Received	Dups	Lost	Hops	Rtmrtrc	ETX	Churn	Beacon Interval	Reboots	CPU Power	LPM Power	Liste
1.1	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	
2.2	9	0	0	1.000	594.333	16....	0	8 min, 00 sec	0	0.373	0.152	
3.3	9	0	0	1.000	595.333	16....	0	8 min, 29 sec	0	0.367	0.152	
4.4	9	0	0	1.000	446.222	16....	0	8 min, 58 sec	0	0.369	0.152	
5.5	8	0	0	1.000	606.375	16....	0	6 min, 49 sec	0	0.383	0.152	
6.6	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	
7.7	9	0	0	2.000	495.222	16....	0	8 min, 58 sec	0	0.367	0.152	
8.8	8	0	0	2.000	849.125	31....	0	6 min, 08 sec	0	0.370	0.152	
9.9	9	0	0	2.000	515.333	16....	0	8 min, 00 sec	0	0.385	0.152	
10.10	9	0	0	2.000	494.667	16....	0	8 min, 00 sec	0	0.363	0.153	
11.11	8	0	0	2.000	498.375	16....	0	6 min, 16 sec	0	0.383	0.152	
12.12	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	
13.13	9	0	0	1.000	256.000	16....	0	2 min, 34 sec	0	0.431	0.150	
Avg	8.700	0.000	0.000	1.500	535.099	17....	0.000	7 min, 13 sec	0.000	0.379	0.152	

Figure 4.1.1b Node information of decreased packet dropping stimulation.

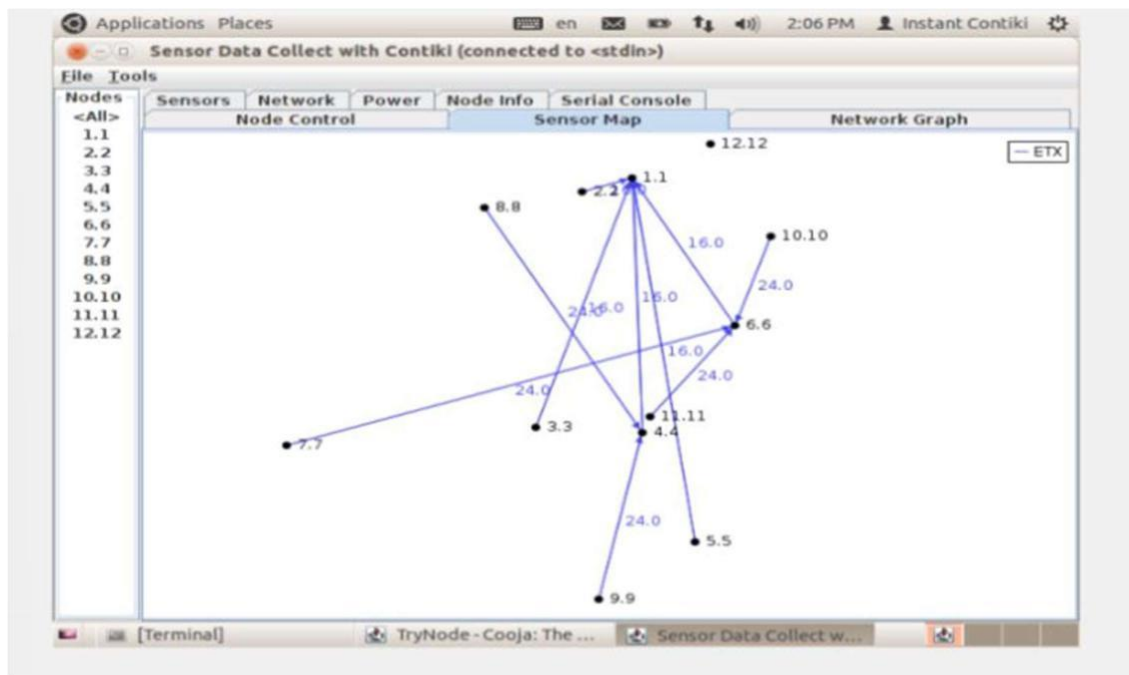


Figure 4.1.1c Sensor map of no Packet dropping stimulation.

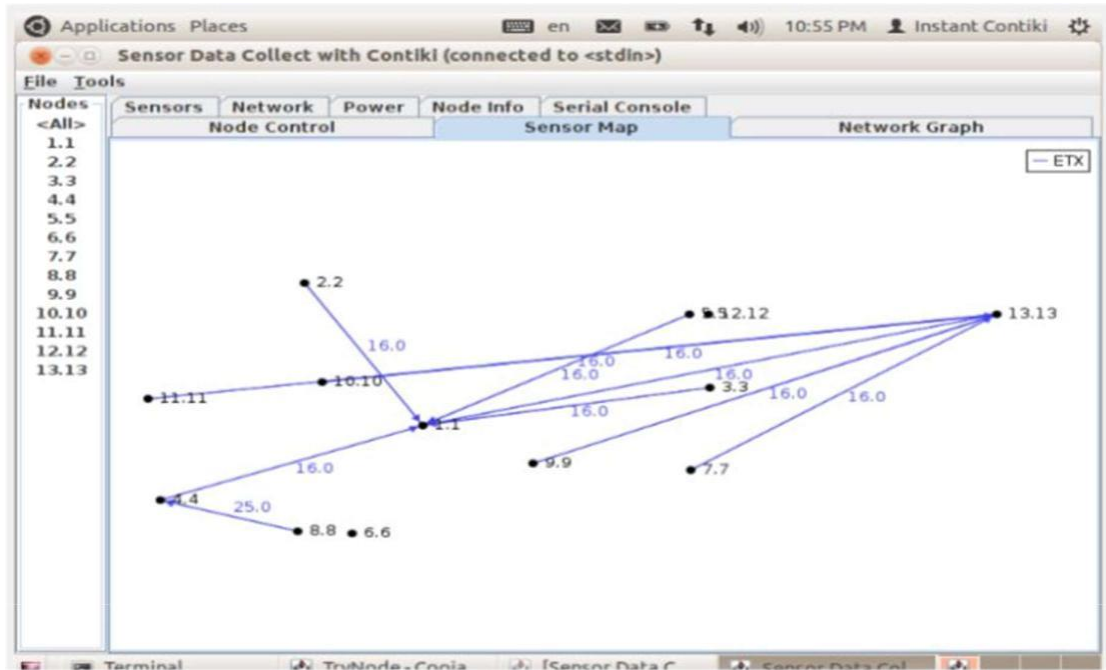


Figure 4.1.Id Sensor map of decreased Packet drop stimulation.

So as we can clearly see from the above figures we not only have change in packed received but also comparing the sensor maps of both simulations we can see in figure 4.1.1c we have an ideal way in which the packets are being sent but in figure we can see that the affected malicious node 13 has disturbed the entire process and has pulled a lot of packets to itself and has affected the entire process.

4.1.2 Comparing the Average Power Consumption Graphs:

Here we will compare Average Power Consumption Graphs of the two no attack and decreased Packet drop attack and if we find any increase in power due to mode 13 then that means the decreased Packet drop attack has taken place.

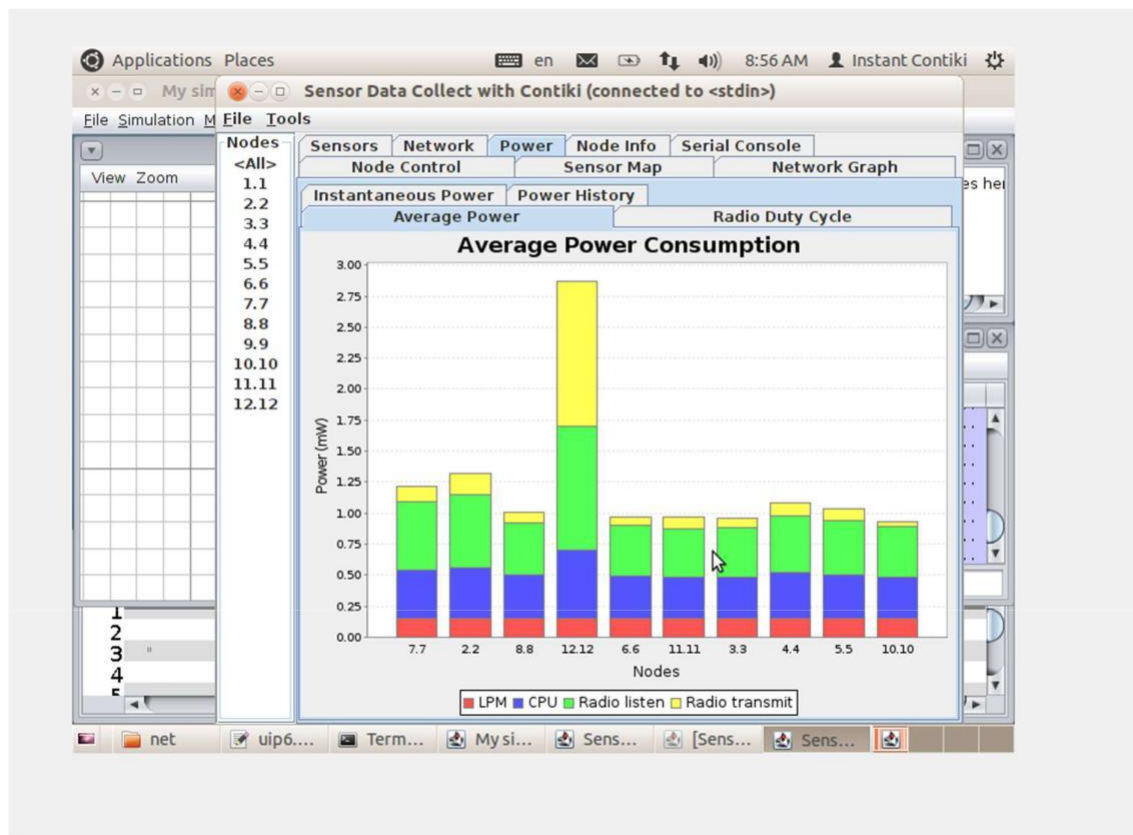


Figure 4.1.2a and c Average Power Consumption Graph of Packet dropping attack



On comparing the above figure 4.1.2a and b we can clearly see how mode 13 when replaces mode 6 and stimulation is run there is Packet drop attack implemented as we can see the average range of both graphs is changed and mode 13 has consumed the highest power disturbing the entire process of packed sending. Hence we can say decreased Packet drop attack has been implemented.

4.2 Algorithm For Detecting and removing packet dropping.

Detecting and removing packet dropping in a network is a complex task that requires a deep understanding of the network architecture and the causes of packet loss. There are several algorithms and techniques that can be used to detect and remove packet dropping in a network, and the choice of algorithm depends on the specific characteristics of the network and the causes of packet loss.

Here is a high-level algorithm that can be used to detect and remove packet dropping in a network:

1. Identify the source and destination of the packets: The first step is to identify the source and destination of the packets that are being dropped. This can be done by analyzing the network traffic and identifying the IP addresses of the source and destination hosts.
2. Measure the packet loss rate: Once the source and destination of the packets are identified, the next step is to measure the packet loss rate. This can be done by monitoring the network traffic and counting the number of packets that are lost between the source and destination.

3. Determine the cause of packet loss: After measuring the packet loss rate, the next step is to determine the cause of packet loss. There are several reasons why packets can be dropped, including congestion, network errors, and faulty hardware.

4. Implement a congestion control algorithm: If the cause of packet loss is congestion, then a congestion control algorithm can be implemented to reduce the amount of traffic on the network and prevent packets from being dropped. This can be done by using techniques such as traffic shaping, quality of service (QoS) policies, and congestion avoidance algorithms.

5. Fix network errors: If the cause of packet loss is network errors, such as faulty cables or routers, then the network errors should be fixed to prevent further packet loss.

6. Replace faulty hardware: If the cause of packet loss is faulty hardware, such as a faulty network card or router, then the faulty hardware should be replaced to prevent further packet loss.

7. Monitor the network: After implementing the appropriate algorithm to detect and remove packet dropping, the network should be monitored to ensure that the packet loss rate is reduced and the network is functioning correctly.

Overall, detecting and removing packet dropping in a network requires a combination of monitoring tools, analysis techniques, and network management strategies. It's important to have a deep understanding of the network architecture and the causes of packet loss to effectively detect and remove packet dropping in a network.

Applications Places en 8:55 AM Instant Contiki

My sim Sensor Data Collect with Contiki (connected to <stdin>)

File Simulation M File Tools

Nodes

<All>

1.1
2.2
3.3
4.4
5.5
6.6
7.7
8.8
9.9
10.10
11.11
12.12

View Zoom

1
2
3
4

Node Control				Sensor Map				Network Graph			
Node	Received	Dups	Lost	Hops	Rtmetric	ETX	Churn	Beacon Interval	Reboots	CP	
1.1	0	0	0	0.000	0.000	0.000	0				
2.2	5	0	0	2.000	584.200	26....	0	4 min, 35 sec	0		
3.3	6	0	0	2.000	776.667	25....	0	6 min, 11 sec	0		
4.4	5	0	0	3.000	1093.8...	44....	0	4 min, 22 sec	0		
5.5	5	0	0	4.000	1110.0...	52....	0	5 min, 40 sec	0		
6.6	6	0	0	1.000	497.000	16....	0	6 min, 11 sec	0		
7.7	5	0	0	1.000	402.600	16....	0	5 min, 27 sec	0		
8.8	5	0	0	2.000	990.800	35....	0	5 min, 40 sec	0		
9.9	0	0	0	0.000	0.000	0.000	0		0		
10.10	5	0	0	4.000	1117.2...	52....	0	6 min, 59 sec	0		
11.11	6	0	0	3.000	1044.5...	42....	0	5 min, 27 sec	0		
12.12	6	0	0	3.000	896.167	40....	0	3 min, 39 sec	0		
Avg	5.400	0.000	0.000	2.500	851.293	35....	0.000	5 min, 25 sec	0.000		

net uip6... Term... Mysi... Sens... Sens... Sens...

4.2.2 Detecting algorithm for Packet dropping attack in cooja.

Detecting and removing packet dropping attacks in Cooja can be achieved using a combination of techniques, including analyzing network traffic and implementing security measures.

Here is a high-level algorithm for detecting and removing packet dropping attacks in Cooja:

1. **Monitor network traffic:** The first step is to monitor the network traffic and analyze the pattern of packet drops. Cooja provides a built-in network analyzer tool that can be used to capture and analyze network traffic.
2. **Identify the source of the attack:** Once the packet dropping pattern is identified, the next step is to determine the source of the attack. This can be achieved by analyzing the network topology and identifying the nodes that are responsible for the packet drops.
3. **Implement security measures:** After identifying the source of the attack, implement security measures to prevent future attacks. This can include implementing access control measures, such as restricting access to critical network resources, or deploying intrusion detection and prevention systems to detect and prevent attacks.
4. **Monitor network traffic:** Continue to monitor the network traffic to ensure that the security measures are effective and to detect any new packet dropping patterns.
5. **Take corrective actions:** If the security measures are not effective in preventing packet drops, take corrective actions, such as blocking traffic from the source of the attack or deploying additional security measures.
6. **Document the incident:** Document the incident and share the findings with the relevant stakeholders, including network administrators and security personnel.

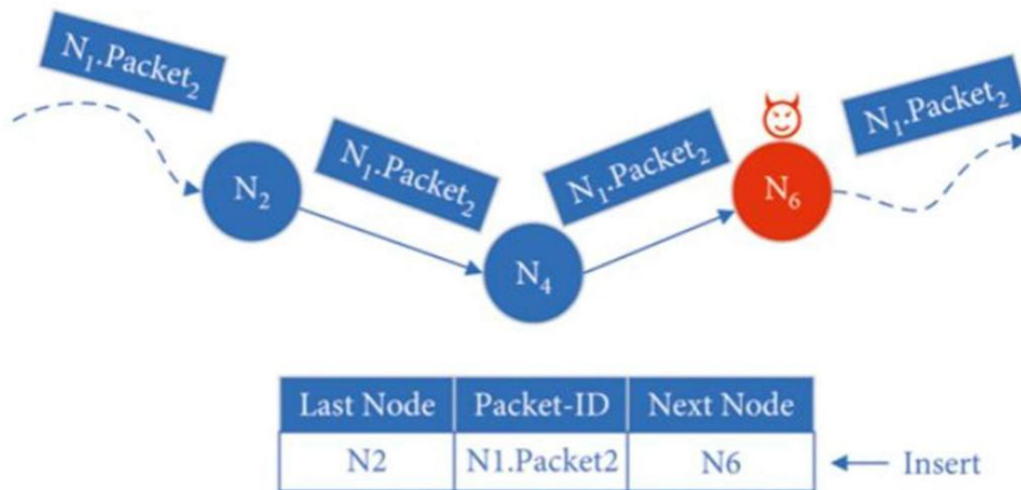
Chapter-5 Conclusions

In conclusion, analyzing the impact of packet dropping attacks in IoT using Cooja can help identify potential security vulnerabilities and provide insights into the effectiveness of current security measures. Packet dropping attacks can significantly impact the performance and reliability of IoT networks, leading to data loss, service disruption, and potentially compromising the security of the network. Through analysis and testing in Cooja, IoT system administrators and security personnel can gain a better understanding of the potential impact of packet dropping attacks and develop strategies to prevent, detect, and mitigate these attacks. It is important to continuously monitor network traffic and implement effective security measures to protect IoT systems from potential packet dropping attacks and other security threats.

Stopping the impact of packet dropping attacks in IoT using Cooja involves implementing a set of security measures to prevent or minimize the effects of such attacks. Here are some steps that can be taken to stop the impact of packet dropping attacks in IoT using Cooja:

1. Use reliable communication protocols: Select communication protocols that are reliable and can handle packet drops effectively. For instance, protocols like TCP can retransmit lost packets, reducing the impact of packet drops.
2. Implement security measures: Implement security measures to protect IoT devices from attacks. This can include encrypting data to prevent unauthorized access, implementing access control measures, and using firewalls to block unauthorized traffic.
3. Monitor network traffic: Continuously monitor network traffic to detect any unusual behavior or patterns, such as packet drops. This can be done using built-in network analysis tools in Cooja.

4. Implement redundancy: Use redundancy to prevent the loss of critical data. For instance, sending multiple copies of data packets to ensure that at least one copy is received correctly.
5. Implement error detection and correction: Implement error detection and correction mechanisms to detect and correct errors in data packets, including packet drops.
6. Perform regular updates: Regularly update the firmware and software of IoT devices to ensure that they are protected against known vulnerabilities and security threats.
7. Educate users: Educate users on the importance of security and how to identify and prevent attacks, such as packet drops.



During the transmission of the packet , receives the packet from and forwards it to . To record this forwarding behavior, inserts a record into its PFRT. Besides, malicious nodes may not update their PFRTs because they drop packets instead of forwarding them.

5.2 FUTURE SCOPE

The impact of packet dropping attacks in IoT using Cooja is likely to continue to be a significant challenge in the future, as IoT networks become increasingly complex and pervasive. Here are some possible future scopes of the impact of packet dropping attacks in IoT using Cooja:

1. Increased sophistication of attacks: Attackers are likely to develop more sophisticated techniques to evade detection and target IoT networks. This may include the use of advanced malware, social engineering tactics, and zero-day exploits.
2. Growth of IoT networks: The growth of IoT networks is expected to continue, with more devices being added to the network. This will increase the attack surface and make it more challenging to detect and prevent packet dropping attacks.

3. Emergence of new attack vectors: As new IoT technologies emerge, new attack vectors may also emerge. For instance, attacks on IoT devices that rely on machine learning algorithms could become more prevalent.

4. Importance of data privacy: With the growing amount of data generated by IoT devices, data privacy is becoming increasingly important. Attackers may target IoT networks to gain access to sensitive data, such as personal information, financial data, or confidential business information.

5. Adoption of new security technologies: As the threat landscape evolves, new security technologies will need to be developed and adopted to protect IoT networks against packet dropping attacks. This may include the use of advanced encryption methods, behavioral analysis, and machine learning algorithms.

In summary, the impact of packet dropping attacks in IoT using Cooja is expected to continue to be a significant challenge in the future. It is essential to develop and implement robust security measures to protect IoT networks against such attacks and stay up-to-date with the latest security technologies and best practices.

REFERENCES

1. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015.
2. J. Jiang, Y. Shi, and H. Zhang. Impact analysis of packet dropping attacks on healthcare IoT systems. *IEEE Access*, 6:77364–77375, 2018.
3. Y. Shi, J. Jiang, Y. Ren, and Y. Zhang. Impact analysis of packet dropping attacks on smart grid systems. *IEEE Transactions on Industrial Informatics*, 14(7):3084–3094, 2018.
4. X. Chen, S. Tang, and H. Li. Impact analysis of packet dropping attacks on industrial control systems. *IEEE Transactions on Industrial Electronics*, 66(9):7204–7214, 2019.
5. L. Zhu, Z. Lu, and J. Wang. Impact analysis of packet dropping attacks on an IoT-based traffic monitoring system. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):934–943, 2018.
6. S. Tavakoli and H. R. Rabiee. Impact of denial-of-service attacks on IoT networks: A review. *IEEE Communications Magazine*, 56(8):68–74, 2018.
7. S. Zeadally, R. Hunt, Y. Chen, A. Irwin, and M. Hassan. Internet of things (IoT) security: A review. *Journal of Network and Computer Applications*, 88:10–28, 2017.
8. P. Xi and P. Liu. Security issues and solutions in the internet of things. *IEEE Internet of Things Journal*, 1(4):358–369, 2014.
9. A. Ullah, M. A. Ullah, S. H. Khan, and W. Saleem. Security in internet of things (IoT): Challenges and solutions. In *2016 International Conference on Frontiers of Information Technology (FIT)*, pages 145–150, 2016.

10. C. Ren, Y. Zhang, L. Yu, and J. Wan. Anomaly detection for industrial IoT: A survey. *IEEE Internet of Things Journal*, 5(1):392–403, 2018.
11. H. Farooq and A. Gani. A comprehensive study on internet of things (IoT) security. *Journal of Network and Computer Applications*, 84:10–28, 2017.
12. S. J. Hasan, M. T. Ahmed, and N. H. Tran. A survey on internet of things architectures. *Journal of Grid Computing*, 17(4):487–510, 2019.
13. R. A. Shaikh, S. Haider, and S. Jameel. Security challenges in the internet of things: A comprehensive study. *Journal of Ambient Intelligence and Humanized Computing*, 10(3):997–1015, 2019.
14. J. Yu, L. Zhang, H. Wang, X. Sun, and Y. Zhang. A survey of security in the internet of things. *IEEE Internet of Things Journal*, 4(5):1250–1258, 2017.
15. N. Abbas, Y. Zhang, and Y. Chen. A survey of security in industrial wireless sensor networks. *IEEE Communications Sur*