ANALYSIS OF BEHAVIORAL CHARACTERISTICS OF JAMMERS TO DETECT MALICIOUS NODES IN MOBILE ADHOC NETWORKS

A Project Report

Submitted to the Faculty of Engineering of
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA, KAKINADA

In partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING

By

S. B. MUKESH
(19481A05M9)

THABASSUM
(19481A05N3)

V. MOHAN SAI
(20485A0523)

V. MANOJ
(19481A05O6)

Under the guidance of
Dr. ADILAKSHMI YANAM, M. Tech, Ph.D,
Professor of CSE Department

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRIRAO KNOWLEDGE VILLAGE
GUDLAVALLERU – 521356
ANDHRA PRADESH

2022-2023

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRI RAO KNOWLEDGE VILLAGE, GUDLAVALLERU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **"ANALYSIS OF BEHAVIOURAL CHARACTERISTICS OF JAMMERS TO DETECT MALICIOUS NODES IN MOBILE ADHOC NETWORKS"** is a bonafide record of work carried out by **S. B. MUKESH (19481A05M9), THABASSUM (19481A05N3), V. MOHAN SAI (20485A0523) and V. MANOJ (19481A05O6)** under the guidance and supervision in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru Technological University Kakinada, Kakinada during the academic year 2022-23.

Project Guide
(Dr. ADILAKSHMI YANAM)

Head of the Department
(Dr. M. BABU RAO)

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragements crown all the efforts with success.

We would like to express our deep sense of gratitude and sincere thanks to **Dr. Adilakashmi Yanam, Professor**, Department of Computer Science and Engineering for her constant guidance, supervision and motivation in completing the project work.

We feel elated to express our floral gratitude and sincere thanks to **Dr. M. Babu Rao**, Head of the Department, Computer Science and Engineering for his encouragements all the way during analysis of the project. His annotations, insinuations and criticisms are the key behind the successful completion of the project work.

We would like to take this opportunity to thank our beloved principal **Dr. G.V.S.N.R.V Prasad** for providing a great support for us in completing our project and giving us the opportunity for doing project.

Our Special thanks to the faculty of our department and programmers of our computer lab. Finally, we thank our family members, non-teaching staff and our friends, who had directly or indirectly helped and supported us in completing our project in time.

Team members

S. B. Mukesh (19481A05M9)

Thabassum (19481A05N3)

V. Mohan Sai (20485A0523)

V. Manoj (19481A05O6)

INDEX

LIST OF ABBREVATIONS

Abbreviation	Explanation
MANETs	Mobile Ad Hoc Network
DoS	Denial of Service
QoS	Quality of Service
DSDV	Destination-Sequenced Distance Vector
PDR	Packet Delivery Ratio
NS	Network Simulator

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Wireless ADHOC Networks are used to establish a wireless connection between two computing devices without the need for a Wi-Fi access point or router. This network is decentralized and uses omnidirectional communication media, which makes it more vulnerable to certain types of attacks compared to wired networks. Jamming attacks, a subset of denial-of-service (DoS) attacks, involve malicious nodes that intentionally interfere with the network, blocking legitimate communication. To address this issue, the proposed method analyzes various characteristics of nodes, such as packets sent, received, and dropped, at each node. Using the packet delivery ratio and packet drop ratio, the method detects jamming nodes from normal nodes, improving network performance. The network is simulated in NS2 environment.

Keywords - ADHOC, DoS, Jamming attack, and NS2.

CHAPTER – 1

INTRODUCTION

1.1 INTRODUCTION

1.1.1 MANETs and it's Features

MANET stands for Mobile Ad hoc Network, which is a type of wireless network that is formed by a collection of mobile devices or nodes without the need for a centralized infrastructure such as a router or access point.

In MANETs, nodes communicate directly with each other, and each node acts as a router by forwarding data packets to other nodes in the network. This decentralized nature makes MANETs ideal for use in situations where traditional wired or wireless networks are unavailable or impractical, such as in disaster recovery, military operations, or remote areas.

Some of the key features of MANETs include:

1. **Decentralization:** Each node in a MANET is equal, and there is no central point of control. Nodes in the network communicate with each other directly, making the network highly decentralized.
2. **Dynamic Topology:** The topology of a MANET is constantly changing as nodes move in and out of the network. This requires that routing protocols used in MANETs be able to adapt to changing network conditions.
3. **Limited Battery Power:** Mobile devices in a MANET typically have limited battery power, which can affect the performance of the network. Routing protocols used in MANETs must take into account the limited battery life of devices.
4. **Security:** MANETs are vulnerable to various security threats such as eavesdropping, spoofing, and denial-of-service attacks. Therefore, security is a critical concern in the design and deployment of MANETs.
5. **Scalability:** MANETs can be scaled up or down depending on the number of nodes and the amount of traffic they generate. However, the scalability of a MANET can be limited by the available resources such as bandwidth and battery life.
6. **Limited Transmission Range:** Nodes in a MANET typically communicate wirelessly, and their transmission range is limited. As a result, a packet may need to be forwarded through multiple intermediate nodes to reach its destination.
7. **Unpredictable Network Behavior:** Because nodes in a MANET can move around and enter/leave the network at any time, the behavior of the network can be

unpredictable. This can cause fluctuations in network performance and make it difficult to predict how the network will behave in the future.

8. **Bandwidth Constraints:** MANETs often have limited bandwidth, which can cause delays and congestion in the network. This can be exacerbated by the dynamic nature of the network, which can lead to frequent route changes and retransmissions.
9. **Routing Protocols:** Because MANETs lack a fixed infrastructure, routing protocols are critical to ensure that packets are forwarded to their destination. However, designing efficient and effective routing protocols for MANETs can be challenging due to the dynamic topology and other constraints mentioned above.
10. **Resource Constraints:** In addition to limited battery power and bandwidth, nodes in a MANET may also have limited processing power and memory. This can affect the performance of the network and the ability of nodes to execute complex protocols or applications.

Fig 1.1.1.1: Working of MANETs

1.1.2 Applications of MANETs

There are many unique features in MANETs that makes them particularly suitable for scenarios where traditional communication infrastructure is unavailable, damaged, or destroyed. Due to their flexibility and resilience, MANETs have numerous potential applications in various fields, including military communications, disaster response, sensor networks, mobile commerce, intelligent transportation systems, emergency response, collaborative learning, industrial automation, smart grids, healthcare, wildlife tracking, mobile social networking, and many more. Each of these applications presents its own set of challenges and requirements, and designing efficient and effective communication protocols for MANETs is an ongoing area of research and development.

1. **Military Communications:** MANETs can be used in military settings to provide secure and reliable communication between soldiers on the ground, vehicles, and command centers. MANETs are particularly useful in situations where traditional communication infrastructure is unavailable or destroyed.
2. **Disaster Response:** In disaster scenarios, communication infrastructure may be damaged or destroyed, making it difficult for emergency responders to coordinate their efforts. MANETs can provide a resilient and decentralized communication network that can be quickly deployed to help coordinate disaster response efforts.
3. **Sensor Networks:** MANETs can be used to connect a large number of sensors in a distributed network, allowing for the collection of data from remote locations. This data can be used for environmental monitoring, surveillance, or other applications.
4. **Mobile Commerce:** MANETs can be used in mobile commerce applications, such as mobile payment systems and mobile banking, where reliable and secure communication is critical.
5. **Intelligent Transportation Systems:** MANETs can be used in intelligent transportation systems to provide communication between vehicles on the road, as well as between vehicles and roadside infrastructure. This can help improve safety, reduce congestion, and optimize traffic flow.
6. **Emergency Response:** MANETs can be used in emergency response situations, such as search and rescue operations, to provide communication between emergency responders on the ground, in the air, and at command centers.
7. **Collaborative Learning:** MANETs can be used in collaborative learning scenarios where students or researchers need to share information and resources in real-time. MANETs can provide a decentralized and flexible communication network for exchanging knowledge and resources.
8. I**ndustrial Automation:** MANETs can be used in industrial automation applications to provide communication between sensors, actuators, and control systems. This can help improve efficiency, safety, and reliability in industrial settings.
9. **Smart Grids:** MANETs can be used in smart grid applications to provide communication between power generation, distribution, and consumption systems. This can help improve energy efficiency, reduce costs, and increase the reliability of the power grid.
10. **Healthcare:** MANETs can be used in healthcare applications to provide communication between medical devices, clinicians, and patients. This can help

improve the quality of care, reduce errors, and increase the efficiency of healthcare delivery.

11. **Wildlife Tracking:** MANETs can be used in wildlife tracking applications to provide real-time tracking and monitoring of animal movements. This can help improve wildlife conservation efforts, and provide valuable data for ecological research.
12. **Mobile Social Networking:** MANETs can be used in mobile social networking applications to provide communication and networking between mobile users in a decentralized and flexible manner. This can help improve social interactions, enhance communication, and promote social cohesion.

1.1.3 Challenges in MANETs

1. **Limited Bandwidth:** MANETs typically operate on limited bandwidth, which can make it difficult to transmit large amounts of data quickly and reliably.
2. **Security:** Since MANETs operate in a distributed and dynamic environment, it can be challenging to provide secure communication between nodes. Security threats can include eavesdropping, data tampering, and denial of service attacks.
3. **Scalability:** MANETs can become congested or overloaded when the number of nodes in the network grows, which can impact the overall performance of the network.
4. **Mobility:** The mobility of nodes in MANETs can make it difficult to establish and maintain stable communication links. This can result in frequent link failures and network disruptions.
5. **Energy Consumption:** Since nodes in MANETs are often battery-powered, energy consumption is a significant concern. Energy-efficient communication protocols and algorithms are required to prolong the network's lifetime.
6. **Routing:** In MANETs, routing protocols must be able to adapt to the changing network topology and route traffic efficiently between nodes. However, this can be challenging in a dynamic and unpredictable environment.
7. **Quality of Service:** Providing quality of service (QoS) guarantees in MANETs can be challenging due to the limited bandwidth, mobility, and other factors that can impact the performance of the network.
8. **Topology Control:** In order to optimize the performance of a MANET, it is important to control the topology of the network. This can be challenging due to the dynamic nature of the network and the mobility of the nodes.

9. **Network Partitioning:** MANETs can be partitioned into disjoint sub-networks due to the mobility of the nodes or the failure of links. This can result in communication disruptions and the need for efficient network reconfiguration.
10. **Network Management:** MANETs can be difficult to manage due to their distributed nature and the lack of centralized control. Network management functions, such as monitoring and fault detection, must be performed in a decentralized and efficient manner.

1.1.4 Attacks in MANETs

Mobile Ad hoc Networks (MANETs) are susceptible to various security attacks due to their distributed and dynamic nature. These attacks can result in the loss of data, unauthorized access to the network, and disruption of normal network operations. Some common attacks that are possible in MANETs include eavesdropping, packet spoofing, denial of service (DoS), Sybil attack, blackhole attack, wormhole attack, grayhole attack, rushing attack, and jamming attack. These attacks can compromise the confidentiality, integrity, and availability of the network, and pose a serious threat to the security and reliability of MANETs. Therefore, developing effective security mechanisms to prevent these attacks is crucial for the successful deployment and operation of MANETs.

1. **Eavesdropping:** Eavesdropping involves a third-party intercepting and monitoring data transmissions between nodes in the network. This can compromise the confidentiality of the information being transmitted.
2. **Packet Spoofing:** Packet spoofing involves an attacker forging packets to impersonate a legitimate node in the network. This can be used to gain unauthorized access to the network or to launch other types of attacks.
3. **Denial of Service (DoS):** Denial of service attacks involve an attacker flooding the network with traffic or overwhelming a specific node with traffic to disrupt normal network operations.
4. **Sybil Attack:** A Sybil attack involves an attacker creating multiple fake identities in the network to gain control or manipulate the network.
5. **Blackhole Attack:** A blackhole attack involves an attacker dropping all packets that it receives, leading to a loss of data in the network.
6. **Wormhole Attack:** A wormhole attack involves an attacker creating a shortcut between two distant parts of the network, allowing it to intercept and modify data between the nodes.

7. **Grayhole Attack:** A grayhole attack involves an attacker selectively dropping some packets, causing network performance to degrade.
8. **Rushing Attack:** A rushing attack involves an attacker pretending to be the shortest path to a destination node and then dropping packets, causing other nodes to route their traffic through the attacker and degrade network performance.
9. **Jamming Attack:** A jamming attack involves an attacker disrupting wireless communication by transmitting noise or interfering with the wireless signals.

1.1.5 Classification of Routing Protocol

Routing protocols in Mobile Ad hoc Networks (MANETs) can be classified based on several factors such as the type of network, network topology, routing strategy, and routing approach. Here are some common classification categories for routing protocols in MANETs:

1. **Proactive Routing Protocols:** Proactive routing protocols maintain up-to-date routing information for all nodes in the network by continuously exchanging control messages. Examples of proactive routing protocols include Optimized Link State Routing (OLSR), Destination-Sequenced Distance Vector (DSDV), and Fisheye State Routing (FSR).
2. **Reactive Routing Protocols:** Reactive routing protocols establish a route only when it is needed, and the route discovery process is initiated by a source node. Examples of reactive routing protocols include Ad hoc On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR), and Temporally Ordered Routing Algorithm (TORA).
3. **Hybrid Routing Protocols:** Hybrid routing protocols combine elements of both proactive and reactive routing protocols. They maintain a subset of the network topology proactively while establishing routes reactively. Examples of hybrid routing protocols include Zone Routing Protocol (ZRP), and Wireless Routing Protocol (WRP).
4. **Location-Based Routing Protocols:** Location-based routing protocols use the physical location of nodes to establish routes. Examples of location-based routing protocols include Greedy Perimeter Stateless Routing (GPSR), and Geographic Routing Protocol (GRP).
5. **QoS-Based Routing Protocols:** Quality of Service (QoS) routing protocols consider metrics such as bandwidth, delay, and jitter to establish routes that meet certain performance requirements. Examples of QoS-based routing protocols include QoS-

Aware Routing Protocol (QARP), and Resource Reservation Protocol (RSVP).

6. **Power-Aware Routing Protocols:** Power-aware routing protocols consider the energy consumption of nodes to establish energy-efficient routes. Examples of power-aware routing protocols include Energy-Efficient Routing Protocol (EERP), and Battery-Aware Dynamic Source Routing (B-DSR).

DSDV:

Destination-Sequenced Distance Vector (DSDV) is a proactive routing protocol that is commonly used in Mobile Ad hoc Networks (MANETs). DSDV was one of the first routing protocols developed for MANETs and is based on the classic distance-vector algorithm, with several enhancements to improve its performance in mobile networks.

In DSDV, each node maintains a routing table that contains the distance and sequence number of each available destination node. The sequence number is used to ensure the freshness of the routing information, which prevents routing loops and stale routes. The sequence number is incremented each time a node updates its routing information and is broadcasted to its neighbors.

DSDV uses periodic updates to maintain up-to-date routing information throughout the network. Nodes broadcast their routing tables to their neighbors at regular intervals, allowing other nodes to update their routing tables accordingly. This proactive approach ensures that nodes have the most recent routing information, which reduces routing overhead and improves network performance.

One of the main advantages of DSDV is its ability to provide loop-free and stable routing paths, which are crucial for MANETs. However, DSDV has several limitations, such as the high overhead of the periodic updates, which can cause network congestion and energy depletion, especially in large and dynamic networks.

To address these limitations, several variants of DSDV have been developed, such as Enhanced DSDV (E-DSDV) and Location-Aided DSDV (L-DSDV), which incorporate location information and reduce the number of updates by using selective updates.

Overall, DSDV is a reliable and efficient routing protocol for MANETs, especially in small and static networks. However, its performance may be limited in large and dynamic networks, where other routing protocols such as Ad hoc On-Demand Distance Vector (AODV) and Dynamic Source Routing (DSR) may be more suitable.

Here's an example with a figure on how DSDV (Destination-Sequenced Distance-Vector) protocol works. Let's say we have a network with four nodes, A, B, C, and D. Each node is connected to one or more of the other nodes. The goal is to find the shortest path from any source node to any destination node in the network.

In DSDV protocol, each node maintains a routing table that lists the available paths to other nodes in the network. Each entry in the table contains the following information:

- Destination address: the address of the destination node.
- Next hop address: the address of the next node along the path to the destination node.
- Number of hops: the number of hops along the path to the destination node.
- Sequence number: A unique sequence number identifies the latest route to the destination node.

The sequence number is used to ensure that each node has the most up-to-date information about the network. When a node learns about a new route to a destination node, it increments the sequence number for that entry in its routing table. Now let's say that node A wants to send a packet to node D. It checks its routing table and finds that it has a direct link to node C, which in turn has a direct link to node D. Node A forwards the packet to node C, which forwards it to node D. Each node updates its routing table with the new sequence number for the path to node D. If a link in the network fails or a new link is added, the routing tables will be updated accordingly. For example, if the link between node C and node D fails, node C will remove the entry for node D from its routing table. When node A wants to send a packet to node D again, it will find that the only path available is through node B and node C. The Table below shows an example of a routing table for node A in a network with nodes A, B, C, and D.

Table 1.1.5.1: DSDV Routing Table for Node A

Destination	Next Hop	Hop	Sequence Number
A	0.0.0.0	0	1
B	B	1	1
C	B	2	1
D	C	3	1

In this example, node A has a direct link to itself (destination address A) and has learned about node B through its direct link. Node A has also learned about node C and node D through node B. The sequence number for each entry is 1, indicating that these are the most recent routes to the corresponding destination nodes.

1.1.6 Jamming Attack

Jamming attacks are a type of denial-of-service (DoS) attack that are commonly performed in Mobile Ad hoc Networks (MANETs). These attacks aim to disrupt the normal communication between nodes by transmitting radio signals on the same frequency as the legitimate communication signals, resulting in interference and loss of data.

There are three types of jamming attacks that can be performed in MANETs: constant jamming and random jamming.

In a **constant jamming attack**, an attacker continuously emits a jamming signal on a specific frequency, disrupting the communication channel of the mobile ad hoc network (MANET). This type of attack can be highly effective in disrupting the communication between nodes, as it denies access to the communication channel for an extended period, making it difficult for legitimate nodes to transmit or receive data. The jamming attack can also lead to the depletion of the nodes' battery power as they continuously attempt to retransmit data or send acknowledgments, leading to decreased network performance and even network failure.

A **periodic jamming attack** is a type of attack in Mobile Ad hoc Networks (MANETs) where a malicious node repeatedly transmits jamming signals at regular intervals, disrupting communication between other nodes in the network. The goal of this attack is to consume the network resources and prevent legitimate nodes from transmitting data, which can lead to a denial-of-service (DoS) attack.

A **random jamming attack** in Mobile Ad hoc Networks (MANETs) is a type of Denial of Service (DoS) attack where an attacker sends out a high-power signal over a wide range of frequencies to disrupt communication between nodes in the network. The jamming signal can cause interference with legitimate communication signals, preventing nodes from transmitting or receiving data. Unlike a targeted jamming attack, where the attacker aims to disrupt communication between specific nodes or a particular area of the network, a random jamming attack is more indiscriminate, affecting all nodes within the range of the jamming signal.

1.2 OBJECTIVES OF THE PROJECT

The primary objective of the analysis of behavioral characteristics of jammers to detect malicious nodes in Mobile Ad hoc Networks (MANETs) is to develop a mechanism that can effectively identify and isolate malicious nodes that are causing jamming attacks in the network. Some specific objectives of this research could include:

1. To investigate the characteristics and patterns of jamming attacks in MANETs, including the frequency of attacks, the duration of attacks, and the types of nodes that are targeted.
2. To develop a behavioral model for jammers based on their attack patterns and characteristics. This model could include factors such as the number of attacks launched, the frequency of attacks, the duration of attacks, and the target nodes.
3. To develop a detection mechanism that can identify malicious nodes based on their behavior and their deviation from the expected behavioral model. This mechanism could use machine learning algorithms, statistical analysis, or other techniques to detect and isolate malicious nodes.
4. To evaluate the effectiveness of the detection mechanism in detecting malicious nodes and reducing the impact of jamming attacks on MANETs. This could involve simulation studies, experimental evaluations, or both.
5. To explore the applicability of the proposed mechanism in different scenarios, such as different network topologies, different types of jammers, and different attack strategies.

By achieving these objectives, the analysis of behavioral characteristics of jammers to detect malicious nodes in MANETs can contribute to the development of more effective security mechanisms for MANETs, improving the reliability and availability of these networks.

1.3 PROBLEM STATEMENT

The "Analysis of Behavioral Characteristics of Jammers to Detect Malicious Nodes in Mobile ADHOC Networks" project addresses the need for a robust mechanism that can effectively detect and isolate malicious nodes that cause jamming attacks in MANETs. Such attacks pose a significant security threat as they can disrupt communication between nodes and result in the loss of vital data. Traditional security mechanisms such as cryptography and authentication are inadequate to counteract these attacks since they assume a secure network environment. Therefore, the project seeks to develop a behavioral model for jammers based on their attack patterns and characteristics to design a mechanism that can accurately identify and isolate malicious nodes.

CHAPTER – 2

LITERATURE REVIEW

The researchers Y. Adilakshmi et al. (2022) of the paper [1] proposes A new and innovative method for address the problem of nodes that act maliciously in Mobile Ad-hoc Networks (MANET) by employing the algorithms used in Machine Learning, in particular Decision Tree and SVM (Support Vector Machine). One of the most common attacks in MANET is the Blackhole attack, where a malicious node enters the network and drops packets instead of forwarding them. The proposed approach detects attacker nodes using machine learning algorithms and compares the network's performance parameters before and after applying the ML algorithms, including Residual Energy, Throughput, Packet Delivery Ratio, and Average End to End delay. The outcomes demonstrated that the suggested approach is effective for networks in order to detecting malicious nodes and improves its overall efficiency of the network. By improving the security of MANET against malicious attacks, this research contributes to the advancement of network security.

The current study by author Abinaya R. et al. (2021) [1] suggests a digital healthcare system that allows patients to create, compile, and save PHR (Personal Health Records). There is a need for greater focus on cost-effectiveness and faster response times in the public cloud platform. The suggested model for healthcare systems utilizes the publisher-observer pattern, which enables patients to review and amend their personal health records (PHRs) before any computations are performed. The cloud system operates as a backend framework that provides an accessible and transparent environment.

In [4], the authors S. Shrestha et al. (2020) proposed an innovative algorithm called RREP which modifies the information in control packets such as sequence number typically used in the AODV (Ad-Hoc On-Demand Distance Vector) routing protocol. The presented algorithm's performance was evaluated and compared to conventional intrusion detection techniques, resulting in its superiority over them. The paper was published in the International Electrical Engineering Congress (IEEE) and can be a valuable reference for researchers interested in securing Mobile Ad-hoc Networks (MANETs) against blackhole attacks using the RREP algorithm.

The authors Y. Adilakshmi et al. (2019) in reference [5] proposed a method to detect intrusion attacks during the communication between mobile nodes in order to ensure uninterrupted data transmission. In their approach, they selected a monitoring node based on a trust value metric, which is a measure of the node's reliability and reputation. To

further enhance the security of the data content, the authors also used a new secret key generation method for encryption. By using this method, the data transmission was protected from intruders who might try to intercept and access the data without authorization. Overall, the proposed method provides a robust solution for securing data transmission in mobile ad-hoc networks.

The paper [6] presented by Y. Adilakshmi et al. (2019) introduces a CIDS (Cooperative Intrusion Detection System) to improve or increase the level of security Mobile Ad-hoc Networks (MANETs). To avoid computation overhead and network failure, a secondary server is selected to perform intrusion detection. The paper proposes the use of a Modified Ant colony algorithm to determine the optimal secondary server. Intrusion detection is performed by learning the traffic variation between different traffic patterns using a modified Support Vector Machine (SVM) approach. Simulation results using NS2 indicate that CIDS has a better intrusion detection probability than existing works. The proposed system can help in improving the security of MANETs by detecting intrusions in a timely and efficient manner.

The research work presented by Y. Guo et al. in 2019 [7] proposed an incentive-based intrusion detection method aimed at enhancing the level of security in the system by increasing the intrusion detection ratio. A game-theoretic method was utilized to effectively carry out intrusion detection in this approach. The method also introduced a punishment appeal mechanism that further improved the efficiency of intrusion detection. By implementing this approach, the research aimed to improve the security level of the system as a whole in terms of its performance, and the findings demonstrated that this method was effective in achieving this goal.

The paper [8] proposed by S. Bambang et al. (2019) states a method for detecting fake access points in wireless networks by analyzing the Media Access Control and Basic Service Set Identifier addresses in beacons. This approach provides the benefit of being lightweight and simple to implement, making it suitable for use on mobile devices. However, the authors acknowledge that this method has limitations and may not be able to detect fake access points created by advanced attackers who can replicate all the static information of a legitimate access point. While this method may not be foolproof, it still offers a valuable tool in the arsenal of wireless network security and can help detect basic types of fake access points.

In [11], the authors B. Alotaibi et al. (2016) highlight that as per the guidelines of the IEEE 802.11 protocol, there exist only a available data rates and modulation types are restricted

in number. The likelihood is high that transmission rate adaptation algorithm of the attacker's fake access point will use the identical data rate or modulation scheme as the authorized access point, particularly in the case that they are functioning on the identical frequency channel and are located in close proximity. As a result, the fake access point of attacker's may go undetected using the modulation-based detection method.

In some studies, authors have proposed methods for Identifying fraudulent access points by measuring and evaluating the beacon signals received. For example, in Kao et al. (2014) [12], researchers suggest that detecting deviations in the beacon interval can be utilized for the identification of counterfeit access points. However, researchers assumed may be attacker has already succeeded in order to synchronize sequence numbers, clock skew elimination, and copy all static information. The proposed method involves examining numerous beacon frames, the interval between the beacon frames of either the fake access point or the authentic access point will eventually deviate. However, the method by which the attacker can synchronize the sequence number is not described by the authors and prevent the clock skew of fake access point. Furthermore, In the event that the attacker has the ability to precisely control the timing of the bogus access point, they can quickly detect any deviation and adjust the beacon interval accordingly.

In (2013) [13], Jadhav and Patil Jadhav proposed a new technique in order to detect attacks like DDOS, called the OEB (Objective Entropy-Based) method. This technique works by predicting DDOS attacks which has low rate variation in packet transfers between destination and the source nodes is used to learn about the network. The approach finds more effective than detecting DDoS attacks in the traditional methods. However, one of major drawbacks by the specified technique is that intrusion detection may be imprecise due to the slight difference between attack and normal traffic. Compared to other methods, this approach is more likely to produce false positive rate.

The author N. Sufyan et al. (2013) [14] proposed a three-dimensional model for detection of attacks in 802.11b radio networks such as jamming. This model considers three parameters, Packet Delivery Ratio, PW (pulse width) , and signal strength of the signal, which leads to a notable enhancement in precision and better classification of jamming attacks. The authors highlight pulse width, signal strength variation, and Packet Delivery Ratio and were observed to produce results that are consistent with the findings. Model that is presented has shown to be effective in detecting attacks in wireless networks such as jamming attacks and its types.

In their work (2013) [15], Khairnar and Kotecha emphasize the difficulties of deploying

VANETs and mechanisms for routing efficiently. They contrast the results of three protocols for routing, they are AODV, DSR, and GPSR, where they are based on discovery of different gateway and topology updating algorithms. They conduct simulations using SUMO and NS2 simulators to evaluate the protocols based on measures like throughput, loss of packets, packet delivery ratio, and end-to-end delay for different scenarios. The authors analyze the simulation results to provide useful insights for the development and deployment of VANET protocols. Overall, their study highlights the importance of effective routing mechanisms for the successful deployment of VANETs.

In their study, Cheng T et al. (2012) [16] produced a method known as Double Circle Localization (DCL) for addressing an issue on localization of the jammer in wireless networks. The authors considered a scenario where all nodes are static in the nature in the network and deployed randomly and can detect if they are being jammed. For detecting the jammer location, proposed algorithm computes the minimum bounding circle and maximum inscribed circle for the all the compromised nodes convex hull. Algorithm's performance was tested through experiments, evaluating its accuracy, efficiency, and robustness using the free-space propagation model. The results showed, DCL approach performs better compared to other methods, this makes it a potentially effective method for identifying the location of jammers in wireless networks.

Author D. Torrieri et al. (2012) [17] suggests a different approach for locating compromised nodes in a wireless network. Unlike many existing methods that rely on the information about the group of nodes that have been intentionally disrupted (jamming nodes) this method focuses on locating the node that has been breached (compromised node) can be identified by analyzing its spread-spectrum signal transmission with a recognized key. One main advantages of this approach is it does not require any assumptions regarding the propagation model of the signal. This approach can be useful in situations where there is little or no information about the jammer or the jammed nodes.

In the article [18], the authors Han C et al. (2012) present a method of verifying the authenticity of a wireless access point by comparing its information from beacon, like the service set identifier, type of authentication, and type of cipher. They argue that authentication type and cipher type are vendor-specific and added using the firmware of WLAN card, making it difficult to the attackers to copy them. However, some authors have published tutorials on modification of firmware of different WLAN cards, suggesting that this information may not always be reliable. To improve accuracy, the combination of other methods comparison of information such as IP addresses from static beacon

information or identifiers of environment, which are located on remaining layers of the Open Systems Interconnection model. These additional pieces of information are used to enhance the accuracy of identifying fake access points.

The authors of the paper [19] Chumchu P et al. (2011) proposed a method to detect the Man-in-the-Middle (MITM) attacks. They did this by analyzing the information provided in beacon frames such as data rate and modulation type, the transmission rate adaptation algorithm defines it. The researchers claimed that their approach is created by the manufacturers of Wireless LAN cards and that modulation type and the data rate change based on the channel's status, so it's not easy to the attackers for manipulating. The writers presented method is effective in finding out attacks such as MITM by conducting experiments in various environments and under different scenarios.

The paper [20] by Liu Hongbo et al. (2011) presents the Virtual Force Iterative Localization algorithm, so it aims to enhance the accuracy of the Centroid Localization method. VFIL considers the jammed nodes distribution and uses a iterative method called as virtual-force to adjust the estimation of the jammer's location. The study conducted by the authors involves comparing the performance of VFIL with other localization algorithms, such as CL. The results indicate that VFIL performs better than CL when it comes to localization accuracy, particularly in situations where only a small number of nodes are jammed. The findings enhance jammer localization techniques in wireless networks.

Y. Xiang et al. (2011) [21] suggested a novel approach for identifying and projecting Distributed Denial of Service attacks in low rate on a network by utilizing metrics based on entropy. This method measures the generalized entropy between the traffic of normal network and DDoS attack traffic to accurately predict the presence DDoS attacks with a low rate. The authors also compare their method with the traditional Shannon entropy method and explained their presented approach has a better performance. The evaluation of this method is based on two metrics, namely false positive rate and distance gap, and the latter is adjusted to ensure reliable and accurate prediction of DDoS attacks. The study shows that this method can effectively detect and predict low-rate DDoS attacks, which are difficult to detect using traditional techniques.

The authors C. Arackaparambil et al. (2010) [22] present a technique for detecting fake access points by using clock difference, also known as clock skew. The authors explain that they can differentiate between the beacons transmitted by the genuine access point and those emitted by using the timestamp clock skew between beacons for analyzing the fake access. Even though this methodology is dependable in recognizing counterfeit access

points, its implementation may pose a challenge. Furthermore, other researchers have discovered that attackers clock skew can be altered in order to reducing their before creating a fake access point by doing this attackers project them self's as legitimate nodes, rendering this technique less effective.

A innovative strategy for analysing ad hoc networks whether there is a jamming attack performed was presented by A. Hamieh et al. (2009) in reference [23]. Their method relies on analyzing the relationship between the duration of error and the time of correct reception. The concept of this paper is to identify a particular jamming type where the jammer only emits when valid radio activity is detected from its radio hardware. To detect such attacks, the transmission node measures the Error Probability and the Correlation Coefficient between the reception error time and the correct reception time. Provided that the Correlation Coefficient is greater than the relative Error Probability, then the network can be classified as jammed. This method allows for detecting jamming attacks without relying on signal strength or channel characteristics, making it a valuable tool in environments with high levels of noise or interference.

The paper by J. Blumenthal et al. (2007) [24] introduces the Weighted Centroid Localization method, which enhances the traditional Centroid Localization technique by considering the influence of jammed nodes during the localization process. WCL assigns weights to each jammed node and modifies its contribution when determining the network's centroid. These weights are calculated using the estimated distance between both the jammer as well as the impacted nodes, which might be calculated by evaluating the radio signal's strength coming in. If the jammed node is nearer to the jammer, the higher its weight will be. This approach provides a more accurate estimation of the jammer's location than CL, which assumes equal contribution from all nodes in the network.

In their paper [25], Guo and Chiueh (2005) proposed a technique for identifying fraudulent access points through examining the gaps between sequence numbers in beacon frames. The authors found that in usual conditions, the difference in sequence numbers between two consecutive beacons is usually less than or equal to 8. If the difference between sequence numbers is larger than 8, then it indicates the presence of a fake access point. While this method is reliable, attackers can potentially examine the time-based variation in the gap between sequence numbers, predict it, and modifying the beacon interval of a fake access point to avoid triggering sequence number gap significant fluctuations. In addition, they may employ jamming signals or requests for probe to interfere with the mechanism for detecting or counting the legitimate access point.

CHAPTER – 3

PROPOSED METHOD

3.1 METHODOLOGY

Jamming attacks are considered more dangerous than other types of attacks in mobile ad hoc networks for several reasons. Firstly, they can cause significant disruption to communication within the network, which can impact critical services and compromise the security of sensitive data. Jamming attacks can disrupt both control and data packets, leading to increased packet loss, increased latency, and decreased throughput. Secondly, jamming attacks can be carried out by low-powered and relatively inexpensive hardware, which makes them easily accessible to attackers. Unlike other types of attacks that may require specialized skills or knowledge, jamming attacks can be executed by anyone with access to basic radio transmitters. Thirdly, jamming attacks can be difficult to detect and locate, as the attacker can operate from a distance without the need for physical proximity to the target network. This makes it challenging for network administrators to identify the source of the attack and take appropriate measures to mitigate the threat. Finally, jamming attacks can be used to create a denial of service (DoS) situation, effectively rendering the network unusable. This can have significant consequences in critical applications such as military, emergency response, and healthcare. Overall, the ease of execution, low cost, and potential for significant damage make jamming attacks a serious threat to mobile ad hoc networks.

When a node in a network is attacked by a jamming attack, it can face several problems. Firstly, the node may experience a significant decrease in the quality of service, as the jamming signal disrupts the communication channel and interferes with the normal transmission of data. This can result in the node being unable to send or receive packets, leading to communication failures and delays.

Secondly, the node may also experience a significant increase in power consumption, as it tries to compensate for the lost communication by repeatedly sending packets, which in turn increases the load on the battery. This can lead to the node draining its battery faster than expected, and may even result in the node being unable to function at all.

Thirdly, the node may also be vulnerable to other attacks while it is busy dealing with the jamming attack. For example, an attacker may exploit the node's vulnerability and launch a different type of attack, such as a routing attack or a spoofing attack, taking advantage of the node's weakened state.

Overall, jamming attacks can cause significant disruption to the network and its nodes, leading to communication failures, increased power consumption, and vulnerability to other attacks.

Jamming attacks can have serious consequences on mobile ad hoc networks (MANETs) and other wireless networks. Firstly, jamming attacks can disrupt the communication between legitimate nodes and prevent them from exchanging data, which can result in the loss of critical information. Secondly, jamming attacks can drain the battery of nodes in the network, as they are forced to retransmit their messages multiple times due to the interference caused by the jammer. Thirdly, jamming attacks can make it easier for an attacker to launch other types of attacks, such as impersonation or interception attacks, as the communication between nodes is compromised. Finally, jamming attacks are difficult to detect and mitigate, as they can be launched from a distance and do not require physical access to the network. These reasons highlight the importance of focusing on jamming attacks and developing effective countermeasures to detect and prevent them in wireless networks.

A **constant jamming attack** is a type of jamming attack in which a malicious node continuously transmits a high-power signal, effectively blocking all communication within the network. This type of attack can be particularly dangerous because it can go undetected for long periods of time and can cause significant disruption to the network. The malicious node may be able to selectively jam specific parts of the network, allowing it to eavesdrop on certain communications while blocking others. Additionally, the constant transmission of high-power signals can drain the batteries of neighboring nodes, leading to a degradation in network performance and potentially causing some nodes to fail altogether.

Detecting and mitigating constant jamming attacks can be challenging because they are often carried out by sophisticated attackers who can adapt their tactics to avoid detection. Traditional security mechanisms, such as cryptography and authentication, may not be effective against these types of attacks. Instead, specialized jamming detection techniques, such as spectrum sensing or statistical anomaly detection, may be required to identify the presence of a constant jammer. Once a jammer is detected, countermeasures such as frequency hopping or power control can be used to mitigate the effects of the attack and restore network performance.

Periodic jamming attacks are a type of jamming attack where the attacker periodically jams the wireless communication channel. In this type of attack, the attacker transmits jamming signals at regular intervals, causing disruptions in the wireless communication between the

nodes in the network. The periodicity of the jamming signals can vary, and the attacker can adjust the frequency and duration of the jamming signals to optimize the impact of the attack.

Periodic jamming attacks are particularly effective in disrupting the communication in mobile ad hoc networks (MANETs), as these networks rely on wireless communication and do not have a fixed infrastructure. The periodic jamming signals can cause nodes in the network to lose synchronization and lead to a breakdown in the routing protocols used to manage communication in the network.

Detection and prevention of periodic jamming attacks are challenging due to the periodic nature of the attack. Traditional intrusion detection systems may not be effective in detecting these attacks as they may not be able to distinguish between the jamming signals and legitimate signals. Some countermeasures that have been proposed to address periodic jamming attacks include adaptive power control, dynamic frequency hopping, and spreading code techniques. These techniques aim to reduce the impact of the jamming signals by minimizing their effect on the communication between the nodes. However, these techniques can also increase the complexity and overhead of the network, making them difficult to implement in practice.

Random jamming attacks are a type of wireless jamming attack that disrupts the communication in a wireless network by transmitting radio frequency signals indiscriminately. In a random jamming attack, the attacker transmits radio signals on the same frequency as the legitimate wireless network, effectively blocking or jamming the communication between nodes in the network. This type of attack is particularly difficult to defend against, as the attacker does not target specific nodes or transmissions, but instead jams all signals in the affected frequency range.

Random jamming attacks can have a severe impact on wireless networks, leading to a loss of critical data and disrupting important services. These attacks are often used by malicious actors to carry out denial-of-service (DoS) attacks, where the goal is to render a network or service unavailable to its users. In addition, random jamming attacks can also be used to eavesdrop on wireless communications by interfering with legitimate transmissions and intercepting the data packets.

Defending against random jamming attacks can be challenging, as traditional security mechanisms such as encryption and authentication may not be effective in preventing these attacks. Some techniques that can be used to mitigate the effects of random jamming attacks include channel hopping, power control, and packet fragmentation. Channel hopping

involves switching between different frequencies to avoid the jammed frequency, while power control adjusts the transmission power of nodes to reduce the impact of the jamming signal. Packet fragmentation involves breaking up large data packets into smaller segments, which can be transmitted more reliably in the presence of a jamming signal.

Overall, random jamming attacks represent a significant threat to wireless networks, and it is important for network administrators to take appropriate measures to defend against these attacks.

Packet delivery ratio (PDR) is an important performance metric in wireless networks, including those that are subject to jamming attacks. In a jamming attack, the attacker transmits interference signals that disrupt or block the legitimate wireless transmissions in the network, resulting in a decreased packet delivery ratio.

The PDR represents the ratio of the number of packets successfully delivered to their intended recipients, to the total number of packets transmitted. A high PDR indicates that a large proportion of the transmitted packets were successfully delivered to their intended recipients, whereas a low PDR indicates that many packets were lost or dropped.

In the context of jamming attacks, a low PDR can indicate that the network is under attack and that a large number of packets are being dropped due to interference. Moreover, if the PDR is consistently low over a period of time, it can be an indication of a persistent jamming attack that requires attention and remedial action.

Therefore, monitoring the PDR is an important aspect of detecting and mitigating jamming attacks in wireless networks. By analyzing the PDR, network administrators can identify the presence of jamming attacks and take appropriate measures to mitigate the attacks and improve the performance of the network.

In a jamming attack, a malicious node broadcasts radio signals that interfere with the communication of other nodes in the network, leading to packet losses and degradation of network performance. **The packet drop ratio** is a measure of the effectiveness of a node in transmitting packets to its intended destination. In the presence of jamming attacks, the packet drop ratio increases significantly, as many packets are lost due to interference.

The packet drop ratio is calculated by dividing the number of packets that were transmitted but not received by the intended recipient by the total number of packets that were transmitted. A high packet drop ratio indicates a high degree of interference in the network and can lead to increased delays and reduced throughput.

Detecting and mitigating the effects of packet drop ratio is crucial in ensuring the reliability and security of the network. Various techniques, such as encryption, frequency hopping,

and power control, can be used to prevent or mitigate the effects of jamming attacks and reduce the packet drop ratio. Additionally, monitoring the packet drop ratio can help detect the presence of jamming attacks and take appropriate measures to mitigate their effects.

Work Flow:

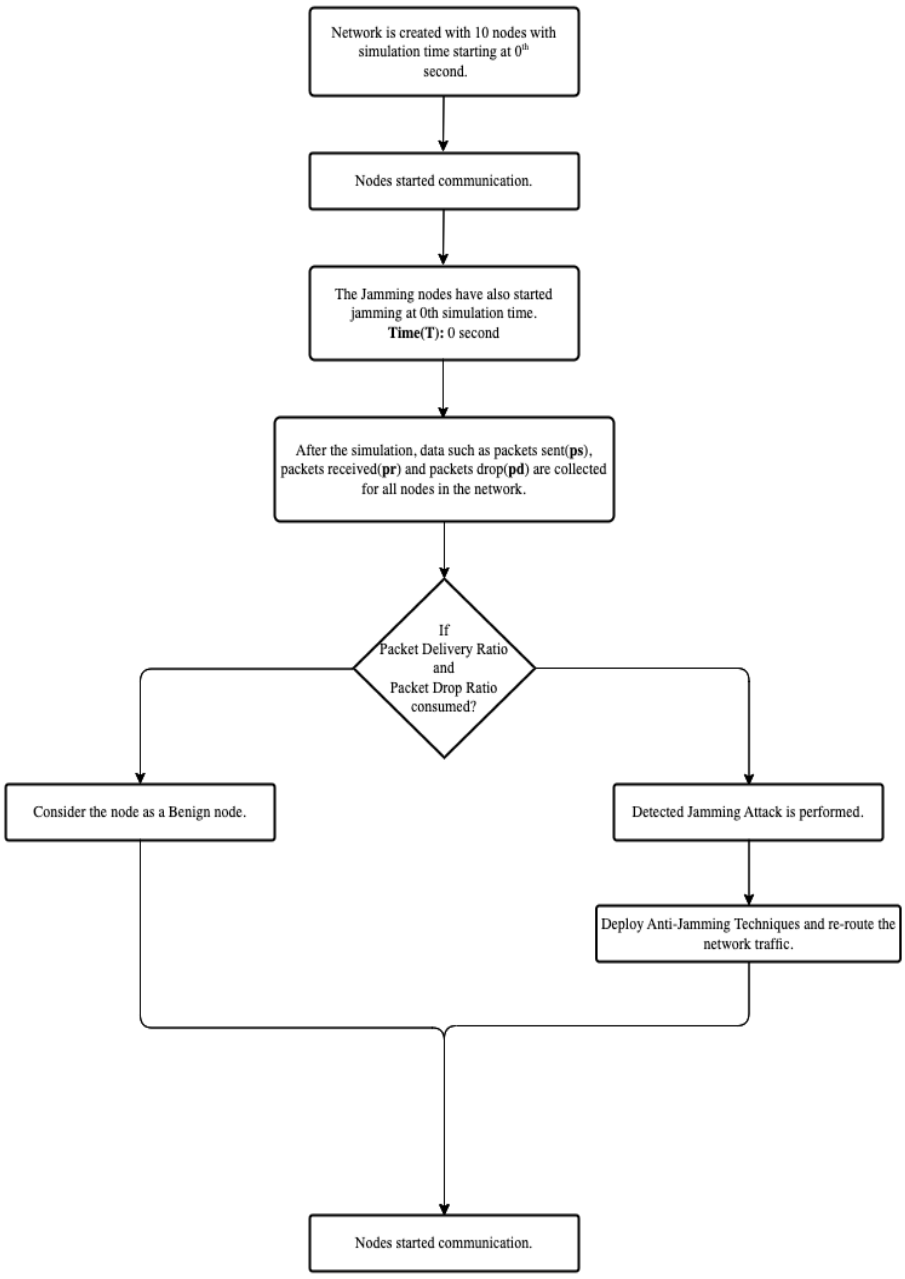

Fig 3.1.1: Workflow of the Project

Initially, a network simulation with 10 nodes is conducted for a duration of 20 seconds, consisting of both benign and malicious nodes. The number of malicious nodes varies from 0 to 5, representing 0% to 50% of the total number of nodes. Communication between nodes starts from the beginning of the simulation without any delay. Once the simulation is completed, trace files are generated, capturing all the communication between any two nodes during the simulation time. Using Python scripts, data such as the number of packets sent, received, and dropped are extracted for each node from the trace files. Graphs are plotted using the extracted data to differentiate between benign and malicious nodes. The packet delivery and drop ratios are calculated from the extracted data. Analyzing these ratios helps in identifying normal and malicious nodes.

3.1.1 Performing Jamming Attack

Jamming attack involves transmitting a radio signal on the same frequency as the wireless communication channel used by the target network. The goal is to flood the channel with interference, disrupting or blocking the communication between nodes. Jamming attacks can be carried out using various techniques, such as continuous wave (CW) jamming, pulse jamming, random jamming, and reactive jamming. In continuous wave jamming, a constant wave is transmitted on the channel, causing interference and disrupting communication. In pulse jamming, the jammer sends high-power pulses on the channel to disrupt the communication. In periodic jamming, the jammer intermittently transmits signals in a periodic interval i.e., being active for a period of time and being inactive for the next period. In random jamming, the jammer intermittently transmits signals at random intervals, making it difficult to detect and counteract the attack. Reactive jamming involves monitoring the network and transmitting the jamming signal only when the jammer detects the presence of legitimate traffic, making it harder to detect and mitigate the attack. Jamming attacks are challenging to defend against since they do not rely on vulnerabilities in the system but on the physical characteristics of wireless communication. Therefore, effective countermeasures require the detection and localization of jammers in real-time, allowing for prompt mitigation of the attack.

3.1.2 Data collection

By simulating the jamming attack in the NS2 environment trace files are generated. These trace files are used for data extraction using python scripts because there’s lots of data in these files. We will be getting the information regarding number of packets send, received and dropped by each node.

3.1.3 Python Scripts

3.1.3.1 Individual Node Graphs

Using information such as number of packets sent, number of packets received and number of packets dropped we plot graphs between number of packets sent, received and dropped on y-axis and simulation time of 20 second on x-axis in order to observe the difference in the nodes i.e., normal nodes or benign nodes.

3.1.3.2 Packet Delivery Ratio

Packet delivery ratio (PDR) is a crucial metric that measures the effectiveness of data transmission in a network by determining The proportion of packets that were effectively transmitted to their intended destination compared to the total number of packets sent. A high PDR suggests an efficient network, whereas a low PDR indicates a problem that needs to be addressed. This metric is especially critical in applications where data transmission reliability is vital, providing a quantitative measure of the network's reliability to ensure that packets are being delivered reliably and efficiently.

$$\text{Packet Delivery Ratio} = \frac{\sum(\textbf{Number of Packets Delivered to Destination})}{\sum(\textbf{Total Number of Packets Sent})} * \mathbf{100}$$

The packet delivery ratio is calculated using the data extracted from the trace file. The data collected provides a representative sample of the actual data, including different percentages of malicious nodes ranging from 0.0 to 0.5, along with their behaviour during constant, periodic, and random attacks. The collected data is presented in the following section.

3.1.3.3 Packet Drop Ratio

Packet drop ratio (PDR) is a crucial metric that calculates the percentage of lost or dropped packets during network transmission. It indicates the efficiency and quality of the network and is particularly significant in applications where data transmission reliability is essential. High PDR can cause poor application performance or complete data loss, while low PDR ensures reliable data transmission and optimal network performance. PDR helps network administrators to identify and troubleshoot network issues promptly, making it an essential metric for measuring network reliability and efficiency.

$$\text{Packet Drop Ratio} = \frac{\sum(\boldsymbol{Number\ of\ Packets\ Not\ Delivered\ to\ the\ Destination})}{\sum(\boldsymbol{Total\ Number\ of\ Packets\ Sent})} * \mathbf{100}$$

The packet drop ratio is calculated using the same data collected for calculating the ratio of packets that are successfully transmitted and received, which includes the quantity of data packets sent. The data that has been collected is utilized to compute the ratio of packets that were dropped.

3.2 IMPLIMENTATION

3.2.1 Ubuntu Installation

You'll need the following things to easily and safely installing Linux alongside Windows:

- A computer that comes preinstalled with Windows 10.
- A USB key (pen drive or USB drive) of at least 4 GB in size and no data on it.
- Internet connection (for downloading Ubuntu ISO image and live USB creating tool). You can do this on any system, not necessarily on the system you are dual booting.
- Optional: External USB disk for making back up of your existing data.

Step 1: Download the required ubuntu version.

Fig 3.2.1.1: Download Ubuntu from official website

Step 2: Create a live USB/disk of Ubuntu

With an assumption that you are using Windows to create the live USB. There are several free applications that allow you to create a live Ubuntu USB. You can use any of these tools. Since I cannot show all of them, I'll go with Rufus.

- Download Rufus for free from its website. It will download a .exe file.
- Plug in your USB. This device is going to be formatted so make sure that you don't have any important data on this USB disk.
- Run the Rufus tool you just downloaded. It automatically identifies the plugged in USB but double check it anyway. Now, browse to the location of the downloaded ISO image and ensure that it uses GPT partitioning scheme and UEFI target system.
- Hit the start button and wait for the process to complete. Your live Linux USB is

ready.

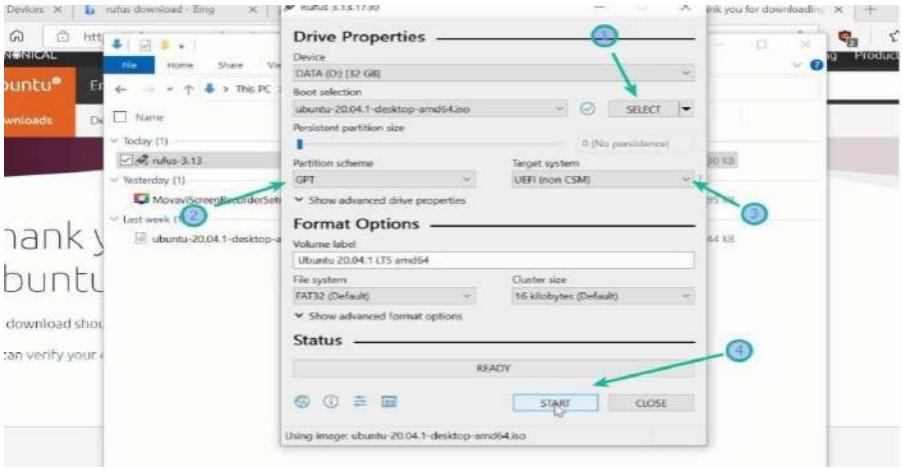

Fig 3.2.1.2: Rufus Process

Step 3: Make some free space on your disk for Ubuntu installation.

- In the Windows menu, search for ‘disk partitions’ and go to ‘Create and format hard disk partitions’.
- In the Disk Management tool, right-click on the drive which you want to partition and select shrink volume.
- If you have just one partition like this, you need to make some free space out of it for Linux. If you have several partitions of considerable size, use any of them except C drive because it may erase the data.
- The 256 GB in my system was already had several partitions from the manufacturer but mainly for backup and other purposes. The main partition was C drive, of around 220 GB, where Windows 10 is installed. In my case, I shrunk the C drive to make some free space for Linux installation.

Step 4: Boot from Live Ubuntu USB

- You created a live Ubuntu USB in the step 2. Plug it in the system. Before you go and boot from the live USB, let’s have a quick word about the infamous secure boot.

Let’s see how to boot from the USB.

- You can go to the boot settings by pressing F2/F10 or F12 at system start time and select to boot from the USB. However, some people find it difficult.
- The longer but easier step is to access the UEFI boot settings from within Windows. In the Windows menu, search for UEFI and then click on ‘Change advanced startup options’:

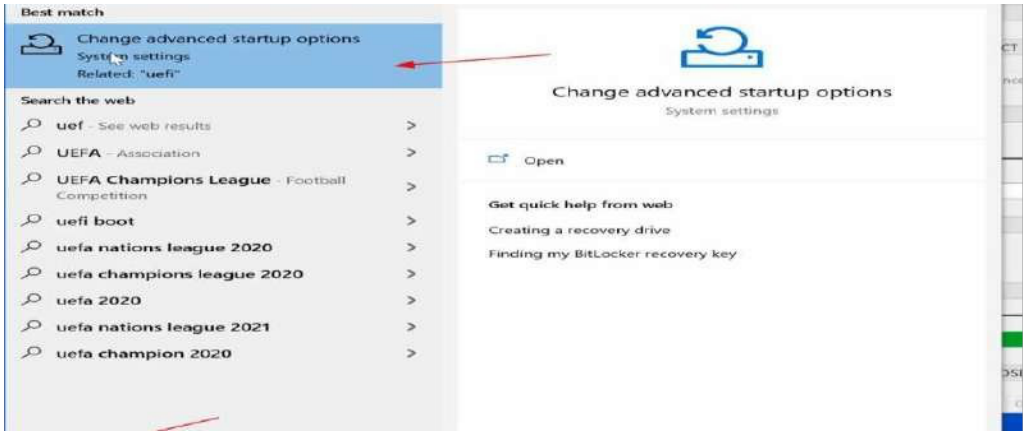

Fig 3.2.1.3: Setting up username

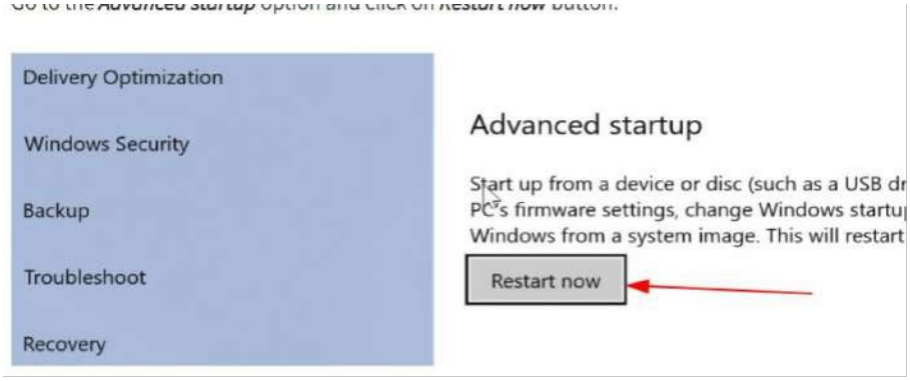

Fig 3.2.1.4: Restarting for Ubuntu

On the next screen, click on 'Use a device':

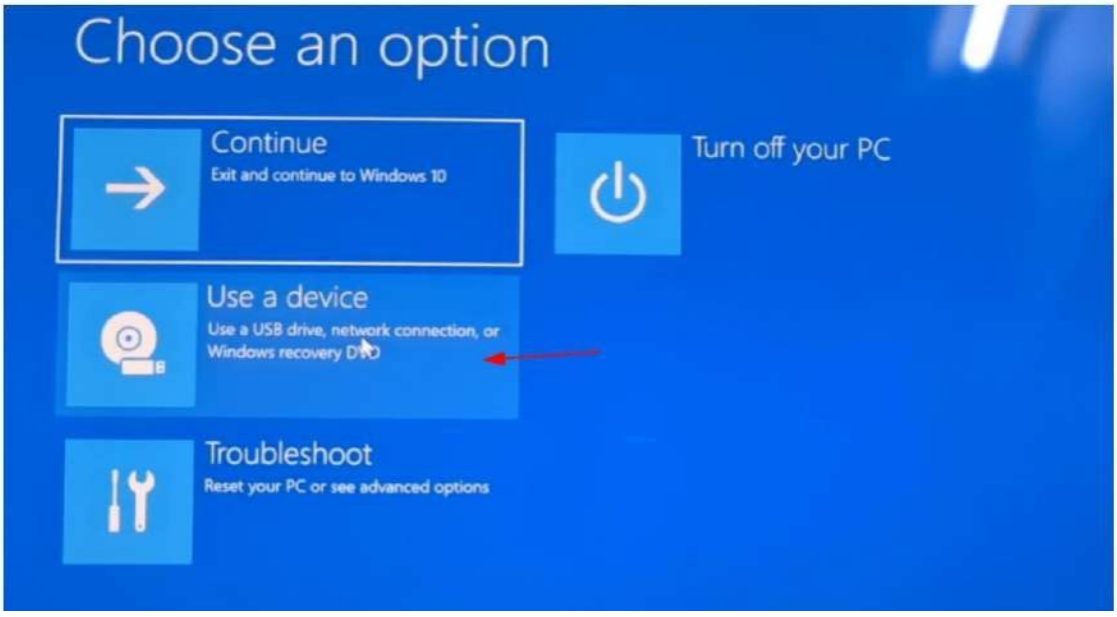

Fig 3.2.1.5: Selecting USB device

- Now it will power off your system and reboot into the disk you chose which should be the live USB disk. You should see a screen like this after a few seconds:

Step 5: Installing Ubuntu along with Windows 10

- Start the installation procedure. The first few steps are simple. You choose the language and keyboard layout.
- On the next screen, choose Normal installation. No need to download updates or install third-party software just yet. You may do it after installation completes.
- Hit continue. It may take some time to go to the next step.

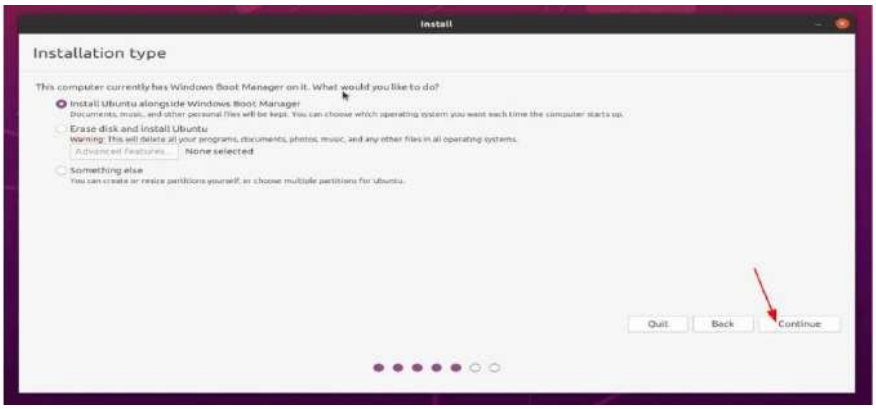

Fig 3.2.1.6: Installing ubuntu alongside windows

- The next screen will give you the option to create a partition for Ubuntu by dragging the divider. You can allocate appropriate disk space to Linux here. Ubuntu will create one partition of the allocated disk space and it will have root with home and a swapfile of 2 GB in size under root itself.
- Next, you'll be asked to enter a username, hostname (computer's name) and a password. Now it's just the matter of waiting. It should take 8-10 minutes to complete the installation.

Fig 3.2.1.7: Preparing for Ubuntu

3.2.2 NS2 Installation

NS2 stands for Network Simulator Version 2. It is an open-source event - driven simulator designed specifically for research in computer communication networks.

Features of NS2:

1. It is a discrete event simulator for networking research.
2. It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, HTTPS and DSR.
3. It simulates wired and wireless network.
4. It is primarily Unix based.
5. Uses TCL as its scripting language.
6. Otcl: Object oriented support
7. Tclcl: C++ and otcl linkage
8. Discrete event scheduler Architecture of NS2

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL.

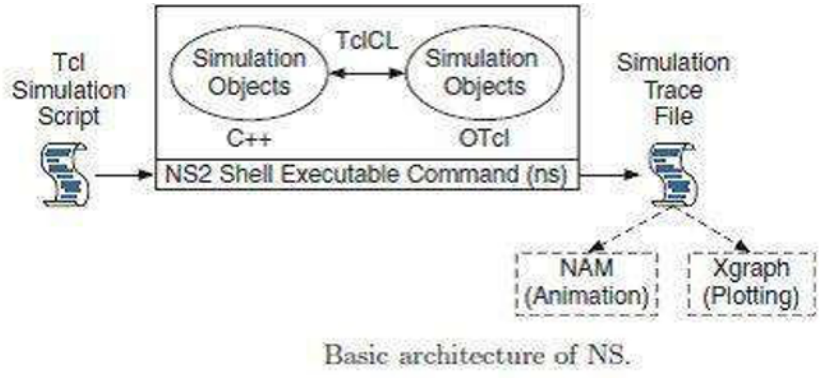

Fig 3.2.2.1: Basic Architecture of NS

Installation of NS2 tool:

For install the NS-2.35, we need the following minimum system requirements.

1. Operating System (OS): ubuntu-14.04 LTS (32 bit)
2. Random Access Memory (RAM): Minimum 2GB
3. Processor: 2.5 GHz and above

Download the NS-2.35:

Download the ns-allinone-2.35.tar.gz package from the following url,

https://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/ns-allinone-2.35.tar.gz/download

To install prerequisites, type and execute the given below command

sudo apt-get install

Change the location by using cd command:

After download the ns-allinone-2.35.tar copy and paste into ubuntu Home location then extract the ns-allinone-2.35.tar file

Execute the cd command:

Open terminal to type the command: cd ns-allinone-2.35

Execute the install command:

Execute the command: sudo ./install

Execute the command cd ns-2.35:

Execute the command cd ns-2.35: cd ns-2.35

Execute the configure command:

Execute the configure command: sudo ./configure

Execute the make command:

Execute the make command: make

3.2.3 Network Simulation

Write all the tcl scripts with all required attributes for simulation of network.

At the end in order to execute the tcl script we had used the python script for automation purpose, by using the automation we can apply different jamming attacks like constant attack, periodic attack and random attack one after the other with new network each time. This helps us in reducing the time required for executing the same tcl file for different attacks and with different number of attackers.

```
set val(chan)                   Channel/WirelessChannel         ;# Channel type
set val(prop)                   Propagation/TwoRayGround        ;# Radio-Propagation model
set val(netif)                  Phy/WirelessPhy                 ;# Network interface type
set val(mac)                    Mac/802_11                      ;# MAC Type
set val(ifq)                    Queue/DropTail/PriQueue         ;# Interface Queue Type
set val(ifqlen)                 1048576;                        ;# 1 MB Queue Length (2^20 bytes)
set val(ll)                     LL                              ;# Link Layer Type
set val(ant)                    Antenna/OmniAntenna             ;# Antenna Model
set val(rp)                     DSDV                            ;# Routing Protocol
set mal_percntg                 [lindex $argv 1]                ;# Initializing the malicious percentage

if {$argc != 6} {                                               ;# Checkiung for invalid command line arguments
        puts "Usage:~\nns $argv0 <node-count> <mal-node-percentage \[0.0-0.9\]> <attack-type\[c/p/r\]> <out-thrput-file> <out-trace-file> <ID>"
        exit 1
} elseif {![string is double [lindex $argv 1]]} {               ;# Checking weather the malicious percentage is in float or number
        puts "The percentage of malicious nodes should be a float"
        exit 1
} elseif {$mal_percntg < 0.0 || $mal_percntg > 0.9} {           ;# Checking weather the malicious node percentage is with in the range
        puts "Invalid malicious node percentage. Please enter a decimal number between 0.1 and 0.9."
        exit 1
}

set val(maxx)                   175
set val(maxy)                   175
set val(nn)                     [lindex $argv 0]                        ;# Initializing the number of nodes
set val(mal_nn)                 [expr {int([lindex $argv 1] * $val(nn))}]       ;# Initializing the number of malicious nodes
set val(ben_nn)                 [expr {$val(nn) - $val(mal_nn)}]        ;# Initializing the number of remaining nodes
set val(attack-type)            [lindex $argv 2]                        ;# Initializing the type of attack
set val(sim_time)               20.0                                    ;# Initializing the simulation time
set val(buff_time)              0.01                                    ;# Initializing the buffer time
set val(scene)                  "scen_$argv0"                           ;# Initializing the file scen file name to scene
set val(conn)                   "conn_$argv0"                           ;# Initializing the file conn file name to scene
set val(events)                 "events_$argv0"                         ;# Initializing the file events file name to scene
set val(mal_time)               0.0
set val(poll_delta)             $val(sim_time)
set val(thrput_file)            [lindex $argv 3]                        ;# Initializing the throughput file name
set val(trace_file)             [lindex $argv 4]                        ;# Initializing the delay file name
set val(ID)                     [lindex $argv 5]                        ;# Initializing the runcount

if { $val(mal_nn) >= $val(nn) } {               # Checking weather the malicious node percentage is less than normal nodes or not
        puts "The malicious nodes are greater or equal to the total number of nodes. This is an invalid configuration."
        exit 1
}

set ns_         [new Simulator]                                 # Creating a simulator object
set topo        [new Topography]                                # Creating a topography object
$topo load_flatgrid $val(maxx) $val(maxy)                       # Creating a grid/place for placing the nodes
create-god $val(nn)                                             # Refer to Doubts.txt
```

Fig 3.2.3.1: Main TCL File

```
for {set i_scen 0} {$i_scen < $val(nn)} {incr i_scen} {
        set node_($i_scen) [$ns_ node]

        $node_($i_scen) set X_ [expr {int( rand() * $val(maxx) )}]
        $node_($i_scen) set Y_ [expr {int( rand() * $val(maxy) )}]
        $node_($i_scen) set Z_ 0.0

        $node_($i_scen) random-motion 0
        $ns_ initial_node_pos $node_($i_scen) 20
        # 20 defines the node size in nam, must adjust it according to your scenario size.

        if {$i_scen >= $val(ben_nn)} {
        # Malicious nodes are created at the last. Colour them red.
                $node_($i_scen) color Red
                $ns_ at 0.0 "$node_($i_scen) color Red"
        }
}
```

Fig 3.2.3.2: Node Creation TCL File

```
# setting up legitimate connections. Each benign node connects only with other benign nodes.
# A random number is generated (either a 0 or 1); depending on which a connection is set up
# between a pair of legitimate nodes.
set legit_conn 0

for {set i 0} {$i < $val(ben_nn)} {incr i} {                    ;# each benign node..
        for {set j 0} {$j < $val(ben_nn)} {incr j} {            ;# connects to benign nodes only
                if {$i != $j} {                                 ;# but, does not connect with itself!
                        set rand_num [expr {int(rand()*2)}]     ;# if rand() >= 0.5 then connection is established
                                                                ;# rand() = 0.6 => 0.6 * 2 => int(1.2) => 1

                        if {$rand_num == 1} {
                                set l_udp_($legit_conn) [new Agent/UDP]
                                $ns_ attach-agent $node_($i) $l_udp_($legit_conn)

                                set l_cbr_($legit_conn) [new Application/Traffic/CBR]
                                $l_cbr_($legit_conn) set rate_ 2e5
                                $l_cbr_($legit_conn) attach-agent $l_udp_($legit_conn)

                                set l_sink_($legit_conn) [new Agent/LossMonitor]
                                $ns_ attach-agent $node_($j) $l_sink_($legit_conn)

                                $ns_ connect $l_udp_($legit_conn) $l_sink_($legit_conn)
                                incr legit_conn
                        }
                }
        }
}

# setting up malicious nodes. A malicious node connects to all nodes (except itself).
# Therefore, at the end of the nested loops below, the mal_conn variable will contain the value:
# malicious nodes * (total nodes - 1)
# This property will become important while implementing periodic and random jamming.
set mal_conn 0

# Please NOTE: If you change the condition of the inner loop (e.g., make the malicious nodes connect
# to only the benign nodes), then you would have to update the initial value of the variable
```

Fig 3.2.3.3: Connections Creation TCL File

```
# The benign nodes are allowed to start at the 0th second. Only the
# start times of the malicious nodes are altered.
for {set i_events 0} {$i_events < $legit_conn} {incr i_events} {
        $ns_ at 0.0 "$l_cbr_($i_events) start"
}

# implementing attacks...
if {$val(attack-type) == "c"} {                                 ;# constant jamming
        for {set i_events 0} {$i_events < $mal_conn} {incr i_events} {
                $ns_ at $val(mal_time) "$m_cbr_($i_events) start"
        }
} elseif {$val(attack-type) == "p"} {                           ;# periodic jamming
        for {set i_events $val(ben_nn)} {$i_events < $val(nn)} {incr i_events} {

                set mal_periods($i_events) [expr {2 + int(rand()*4)}]   ;# The malicious time interval can be between [2, 5]
                set mal_node_status($i_events) 0                        ;# Initially, all malicious nodes are off..

                $ns_ at $val(mal_time) "toggle_periodic_mal_cbr $i_events"      ;# ..but now, we will turn them on
        }
} elseif {$val(attack-type) == "r"} {           ;# random jamming
        for {set i_events $val(ben_nn)} {$i_events < $val(nn)} {incr i_events} {
                set mal_node_status($i_events) 0                ;# Initially, all malicious nodes are off..
                $ns_ at $val(mal_time) "toggle_random_mal_cbr $i_events"        ;# ..but now, we will turn them on
        }
} else {                                                        ;# invalid argument
        puts "Invalid attack type. Valid attack types are:-\n\tc - constant\n\tp - periodic\n\tr - random"
        exit 0
}


# Periodic jamming - Procedure to toggle malicious CBRs.
proc toggle_periodic_mal_cbr {id} {
        global ns_ val mal_node_status mal_periods m_cbr
        set now [$ns_ now]
        set next_time [expr {$now + $mal_periods($id)}]

        # Toggling status of the malicious node using bitwise XOR
        set mal_node_status($id) [expr {$mal_node_status($id) ^ 1}]
        # Number of outgoing connections per malicious node.
        set conns_per_mal_node [expr {$val(nn) - 1}]

        set first_conn [expr {($id - $val(ben_nn)) * $conns_per_mal_node}]
        set last_conn [expr {$first_conn + $conns_per_mal_node}]

        for {set i_events $first_conn} {$i_events < $last_conn} {incr i_events} {
                if {$mal_node_status($id) == 1} {
                        $m_cbr_($i_events) start
                } else {
                        $m_cbr_($i_events) stop
                }
```

Fig 3.2.3.4: Events TCL File

```
from os import system, remove
from time import time
from multiprocessing import Pool
from datetime import timedelta

tcl_script = "wireless_jam.tcl"
attacks = ["c", "p", "r"]               # types of attacks: constant (c), periodic (p), random (r)
out_file_names = ["Thrput_", "Delay_"]      # output file name prefixes of a single run
node_count = 10
run_count = 5


def calc_delay(trace_file, nodes, mal_percentage):
    avg_delay = 0
    sent_time = dict()  # of the form sent_time[pkt_id] = time
    pckts_recvd = 0
    ben_nodes = int(nodes * (1 - mal_percentage))

    try:
        with open(trace_file, "r") as in_fh:
            for line in in_fh:
                split_line = line.split()

                if len(split_line) < 20 or split_line[3] != "AGT":  #Checking conditions for invalid lines using length and AGT
                    continue
                sender_id = int(split_line[13][1:].split(":")[0])
                if sender_id >= ben_nodes:              #Checking weather the sender is within the normal nodes
                    continue
                pkt_id = int(split_line[5])             #Extracting the packet_id
                time = float(split_line[1])             #Extrating the time element from the line

                if line.startswith("s "):               #Checking weather the packet is sent(s) or recieved(r)
                    sent_time[pkt_id] = time
                elif line.startswith("r "):
                    pckts_recvd += 1
                    avg_delay += time - sent_time.pop(pkt_id)       #Calculating the total delay by subtracting the sent time and recieved time

    except KeyError:
        print("Mismatch in packet IDs in the trace file: " + trace_file)
        print("Exiting...")
        exit(1)

    avg_delay /= pckts_recvd                    #Calculating average delay
    remove(trace_file)
    return avg_delay
```

Fig 3.2.3.5: Main Execution Python File

Open Command prompt and redirect to location of tcl file.

By using command to run python file which internally executes tcl scripts:

python3 run_ns_mp

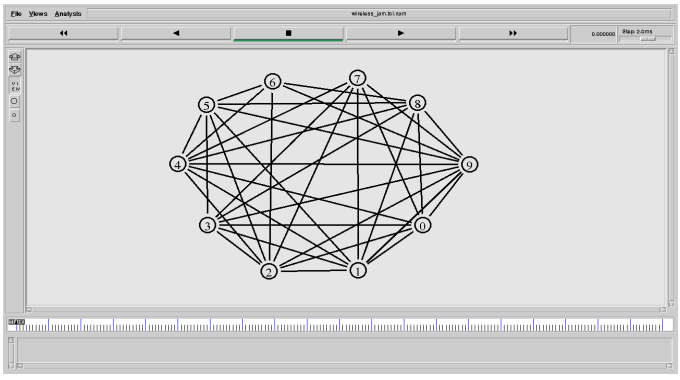

Fig 3.2.3.6: Network Simulation

Once the simulation is done a trace file will be generated which contain all the details of events that occurred in simulated network.

```
s 0.000000000 _0_ AGT  --- 0 cbr 210 [0 0 0 0] ------- [0:0 1:0 32 0] [0] 0 0
s 0.000000000 _0_ AGT  --- 1 cbr 210 [0 0 0 0] ------- [0:1 2:0 32 0] [0] 0 0
s 0.000000000 _0_ AGT  --- 2 cbr 210 [0 0 0 0] ------- [0:2 3:0 32 0] [0] 0 0
s 0.000000000 _0_ AGT  --- 3 cbr 210 [0 0 0 0] ------- [0:3 8:0 32 0] [0] 0 0
s 0.000000000 _1_ AGT  --- 4 cbr 210 [0 0 0 0] ------- [1:1 0:4 32 0] [0] 0 0
s 0.000000000 _1_ AGT  --- 5 cbr 210 [0 0 0 0] ------- [1:2 2:1 32 0] [0] 0 0
s 0.000000000 _1_ AGT  --- 6 cbr 210 [0 0 0 0] ------- [1:3 4:0 32 0] [0] 0 0
s 0.000000000 _1_ AGT  --- 7 cbr 210 [0 0 0 0] ------- [1:4 6:0 32 0] [0] 0 0
s 0.000000000 _1_ AGT  --- 8 cbr 210 [0 0 0 0] ------- [1:5 7:0 32 0] [0] 0 0
s 0.000000000 _2_ AGT  --- 9 cbr 210 [0 0 0 0] ------- [2:2 1:6 32 0] [0] 0 0
s 0.000000000 _2_ AGT  --- 10 cbr 210 [0 0 0 0] ------- [2:3 4:1 32 0] [0] 0 0
s 0.000000000 _2_ AGT  --- 11 cbr 210 [0 0 0 0] ------- [2:4 6:1 32 0] [0] 0 0
s 0.000000000 _2_ AGT  --- 12 cbr 210 [0 0 0 0] ------- [2:5 9:0 32 0] [0] 0 0
s 0.000000000 _3_ AGT  --- 13 cbr 210 [0 0 0 0] ------- [3:1 0:5 32 0] [0] 0 0
s 0.000000000 _3_ AGT  --- 14 cbr 210 [0 0 0 0] ------- [3:2 1:7 32 0] [0] 0 0
s 0.000000000 _3_ AGT  --- 15 cbr 210 [0 0 0 0] ------- [3:3 2:6 32 0] [0] 0 0
s 0.000000000 _3_ AGT  --- 16 cbr 210 [0 0 0 0] ------- [3:4 6:2 32 0] [0] 0 0
s 0.000000000 _3_ AGT  --- 17 cbr 210 [0 0 0 0] ------- [3:5 8:1 32 0] [0] 0 0
s 0.000000000 _3_ AGT  --- 18 cbr 210 [0 0 0 0] ------- [3:6 9:1 32 0] [0] 0 0
s 0.000000000 _4_ AGT  --- 19 cbr 210 [0 0 0 0] ------- [4:2 2:7 32 0] [0] 0 0
s 0.000000000 _4_ AGT  --- 20 cbr 210 [0 0 0 0] ------- [4:3 6:3 32 0] [0] 0 0
s 0.000000000 _4_ AGT  --- 21 cbr 210 [0 0 0 0] ------- [4:4 7:1 32 0] [0] 0 0
s 0.000000000 _4_ AGT  --- 22 cbr 210 [0 0 0 0] ------- [4:5 8:2 32 0] [0] 0 0
s 0.000000000 _5_ AGT  --- 23 cbr 210 [0 0 0 0] ------- [5:0 0:6 32 0] [0] 0 0
s 0.000000000 _5_ AGT  --- 24 cbr 210 [0 0 0 0] ------- [5:1 1:8 32 0] [0] 0 0
s 0.000000000 _5_ AGT  --- 25 cbr 210 [0 0 0 0] ------- [5:2 4:6 32 0] [0] 0 0
s 0.000000000 _6_ AGT  --- 26 cbr 210 [0 0 0 0] ------- [6:4 0:7 32 0] [0] 0 0
s 0.000000000 _6_ AGT  --- 27 cbr 210 [0 0 0 0] ------- [6:5 1:9 32 0] [0] 0 0
s 0.000000000 _6_ AGT  --- 28 cbr 210 [0 0 0 0] ------- [6:6 2:8 32 0] [0] 0 0
```

Fig 3.2.3.7: Generated Trace File

In this way we will be getting multiple trace files for all different attacks with varying number of attackers in the network. Out of the trace files we will extract only characteristics like number of packets sent, number of packets received and number of packets dropped.

3.2.4 Packets Sent, Packet Received and Packet Dropped

From the trace file we will be analyzing the node behaviour by plotting the graphs for packets sent, received and dropped information by using this python script.

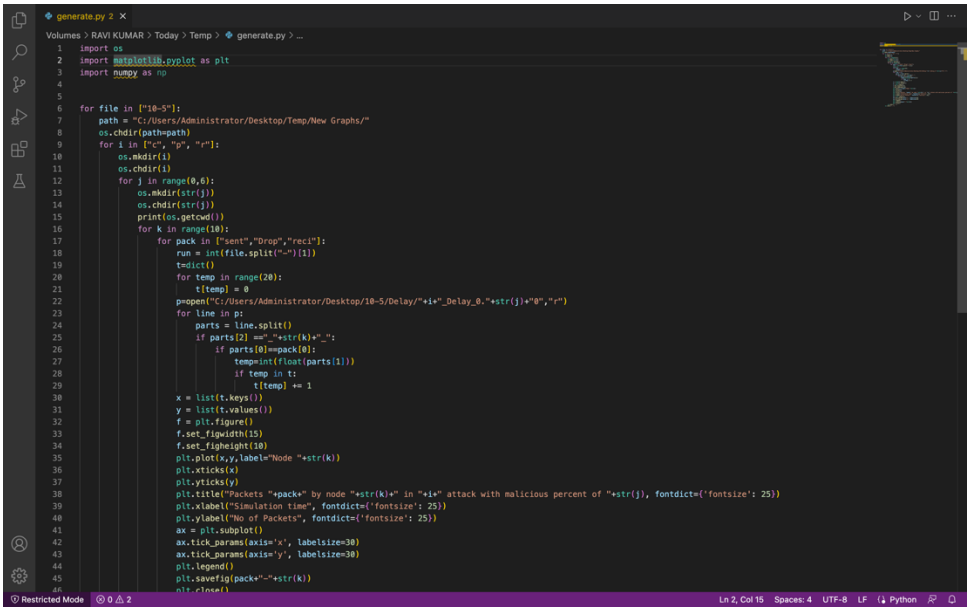

Fig 3.2.4.1: Python file for packets sent, received and dropped graphs

3.2.5 Packet Delivery Ratio and Packet Drop Ratio

The packets sent, packets received and packet dropped information will be noted in an excel using these python scripts.

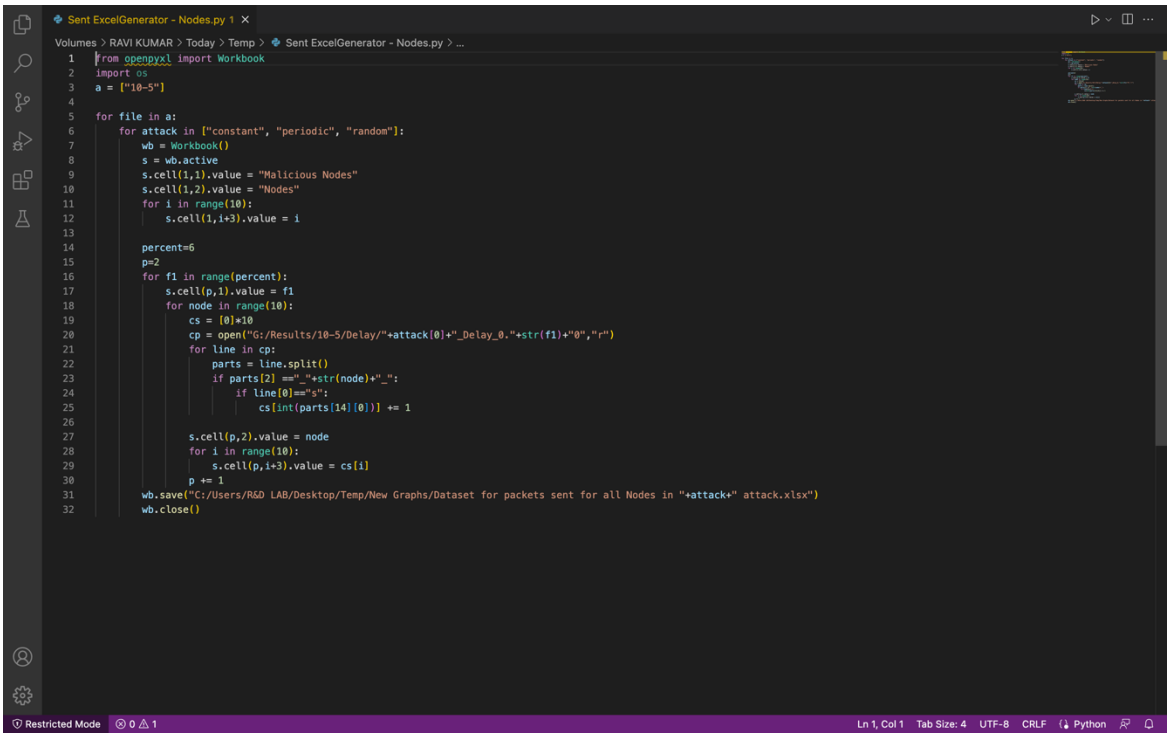

Fig 3.2.5.1: Python file for Packets Sent Excel

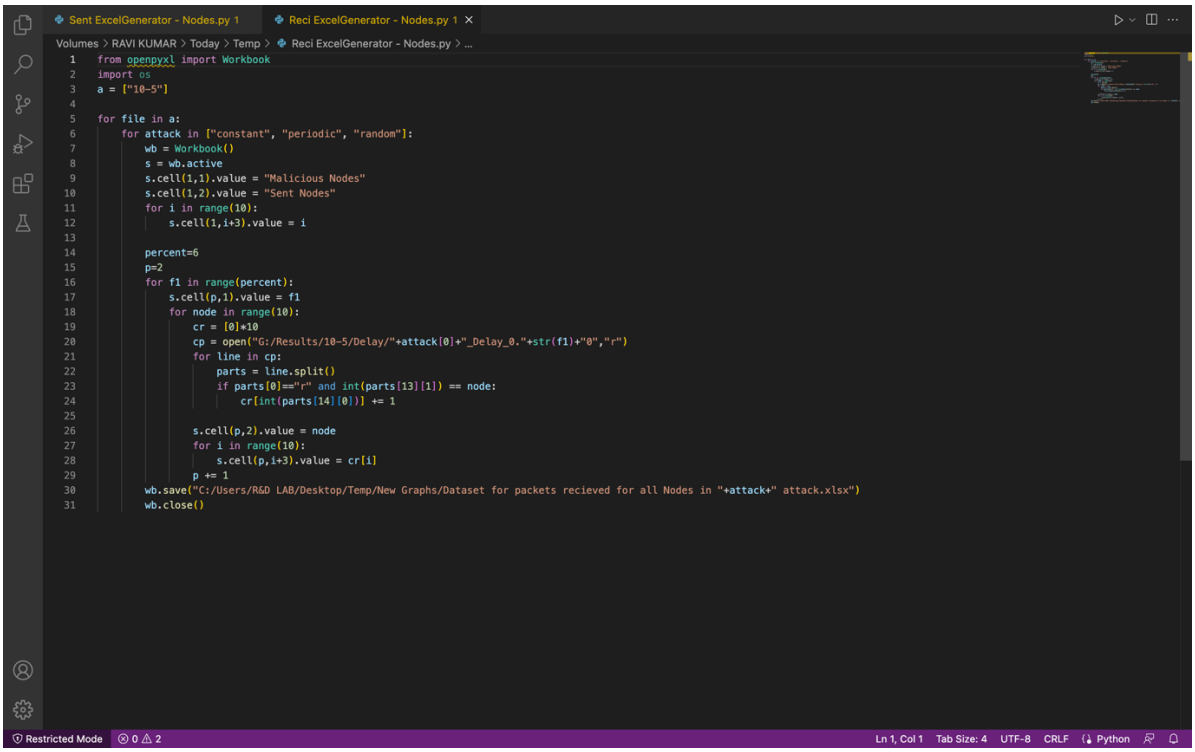

Fig 3.2.5.2: Python file for Packets Received Excel

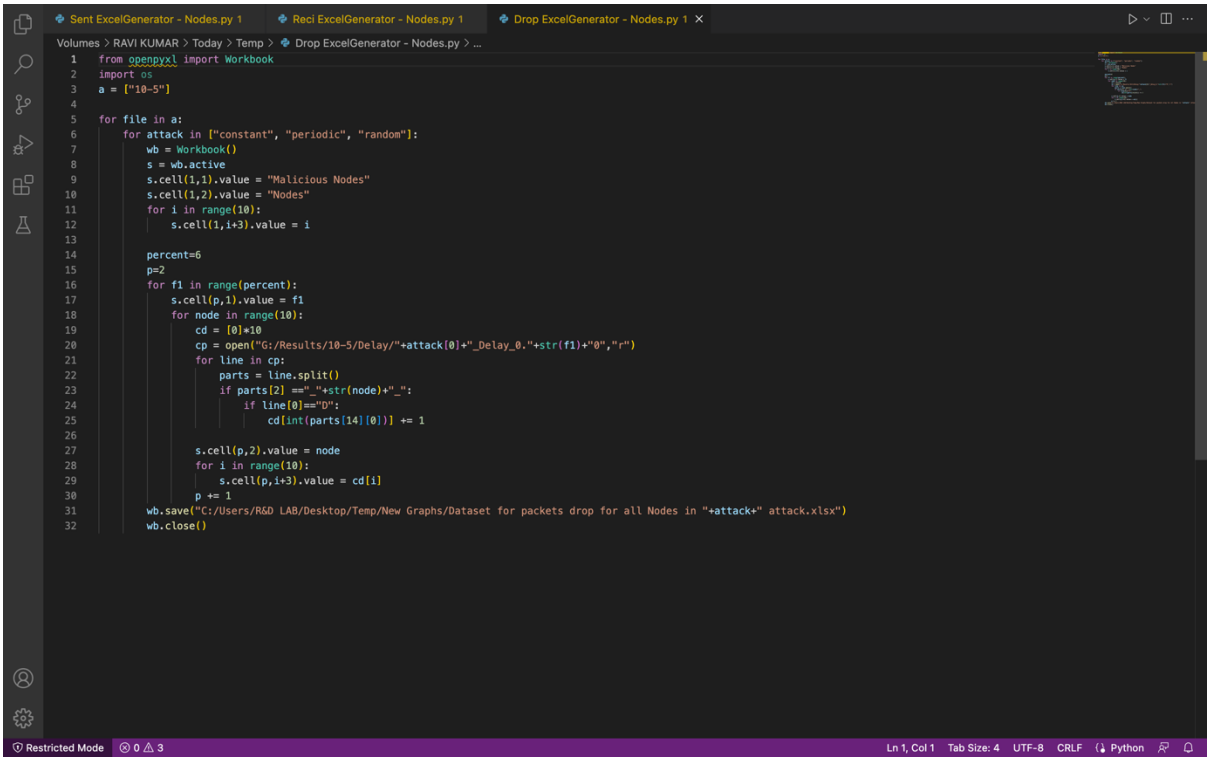

Fig 3.2.5.3: Python file for Packets Dropped Excel

3.3 DATA PREPARATION

Dataset contains the properties of nodes like

1. No of Packets sent by node
2. No of Packets received by node.
3. No of packets dropped by node.
4. Packet Delivery Ratio
5. Packet Drop Ratio

Malicious Nodes	Nodes	0	1	2	3	4	5	6	7	8	9	Total Sent	0	1	2	3	4	5	6	7	8	9	Total Received	Packet Delivery Ratio
0	0	0	2383	2383	2383	0	0	0	0	2383	0	9532	0	189	0	291	0	0	0	0	0	0	480	0.050356693
	1	2383	0	2383	0	2383	0	2383	2383	0	0	11915	232	0	0	0	0	0	157	42	0	0	431	0.036172891
	2	0	2383	0	0	2383	0	2383	0	0	2383	9532	0	136	0	0	0	0	188	0	0	117	441	0.046265212
	3	2383	2383	2383	0	0	0	2383	0	2383	2383	14298	214	88	0	0	0	0	142	0	0	71	515	0.036019024
	4	0	0	2383	0	0	0	2383	2383	2383	0	9532	0	0	0	0	0	0	246	131	0	0	377	0.039550986
	5	2383	2383	0	0	2383	0	0	0	0	0	7149	290	162	0	0	0	0	0	0	0	0	452	0.063225626
	6	2383	2383	2383	2383	2383	2383	0	0	0	0	14298	202	74	0	177	0	12	0	0	0	0	465	0.032522031
	7	2383	0	2383	0	2383	0	0	0	0	0	7149	298	0	289	0	0	0	0	0	0	0	587	0.082109386
	8	2383	2383	2383	2383	0	0	2383	2383	0	2383	16681	156	25	123	128	0	0	78	0	0	7	517	0.030993346
	9	2383	2383	2383	2383	0	2383	2383	2383	0	0	16681	201	77	0	180	0	14	129	14	0	0	615	0.036868293
1	0	0	2383	0	0	2383	0	0	0	2383	0	7149	0	209	0	0	233	0	0	0	0	0	442	0.061826829
	1	2383	0	2383	0	2383	0	0	0	0	0	7149	247	0	186	0	143	0	0	0	0	0	576	0.080570709
	2	0	2383	0	2383	2383	0	0	0	2383	0	9532	0	107	0	0	130	0	0	0	0	0	237	0.024863637
	3	0	2383	0	0	0	0	2383	2383	0	0	7149	0	123	0	0	0	0	173	49	0	0	345	0.048258498
	4	2383	2383	2383	2383	0	0	2383	0	0	0	11915	179	51	98	0	0	0	101	0	0	0	429	0.036005036
	5	0	2383	2383	2383	0	0	2383	0	2383	0	11915	0	64	183	111	0	0	114	0	0	0	472	0.039613932
	6	0	2383	2383	0	2383	0	0	2383	2383	0	11915	0	94	210	0	118	0	0	21	0	0	443	0.037180025
	7	0	0	0	2383	0	0	2383	0	0	0	4766	0	0	0	312	0	0	258	0	0	0	570	0.119597146
	8	2383	2383	2383	2383	2383	2383	2383	0	0	0	16681	177	53	115	22	76	0	103	0	0	0	546	0.032731851
	9	13743	13743	13743	13743	13743	13743	13743	13743	13743	0	123687	397	0	0	0	0	0	0	0	0	0	397	0.003209715
2	0	0	2383	0	0	0	0	2383	2383	0	0	7149	0	16	0	0	0	0	0	0	0	0	16	0.002238075
	1	0	0	2383	0	2383	2383	0	2383	0	0	9532	0	0	357	0	0	171	0	0	0	0	528	0.055392363
	2	0	0	0	2383	0	0	2383	2383	0	0	7149	0	0	0	0	0	0	0	60	0	0	60	0.008392782
	3	0	2383	0	0	0	0	2383	2383	0	0	7149	0	14	0	0	0	0	0	0	0	0	14	0.001958316
	4	0	0	0	2383	0	2383	0	0	0	0	4766	0	0	0	0	0	393	0	0	0	0	393	0.082459085
	5	2383	2383	0	0	0	0	0	2383	0	0	7149	299	169	0	0	0	0	0	0	0	0	468	0.065463701
	6	2383	0	0	0	0	2383	0	2383	0	0	7149	368	0	0	0	0	178	0	0	0	0	546	0.076374318
	7	0	0	2383	2383	0	0	2383	0	0	0	7149	0	0	377	0	0	0	0	0	0	0	377	0.052734648
	8	13743	13743	13743	13743	13743	13743	13743	13743	13743	13743	123687	399	0	0	0	0	0	0	0	0	0	399	0.003225885
	9	13743	13743	13743	13743	13743	13743	13743	13743	13743	0	123687	353	0	0	0	0	0	0	0	0	0	353	0.002853978
3	0	0	2383	2383	0	2383	2383	2383	0	0	0	11915	0	0	201	0	86	12	0	0	0	0	299	0.025094419
	1	0	0	0	0	2383	0	0	0	0	0	2383	0	0	0	0	229	0	0	0	0	0	229	0.096097356
	2	2383	0	0	2383	0	0	2383	0	0	0	7149	220	0	0	0	0	0	36	0	0	0	256	0.035809204
	3	0	2383	2383	0	2383	0	2383	0	0	0	9532	0	0	214	0	99	0	12	0	0	0	325	0.034095678
	4	0	2383	0	0	0	2383	2383	0	0	0	7149	0	0	0	0	0	104	207	0	0	0	311	0.043502588
	5	2383	0	2383	0	2383	0	2383	0	0	0	9532	157	0	162	0	47	0	0	0	0	0	366	0.038396979
	6	0	2383	0	2383	0	0	0	0	0	0	4766	0	145	0	0	0	0	0	0	0	0	145	0.030423836
	7	13743	13743	13743	13743	13743	13743	13743	0	13743	13743	123687	174	0	221	0	0	0	0	0	0	0	395	0.003193545
	8	13743	13743	13743	13743	13743	13743	13743	13743	0	13743	123687	109	0	315	0	0	0	0	0	0	0	424	0.003428008
	9	13743	13743	13743	13743	13743	13743	13743	13743	13743	0	123687	217	0	198	0	0	0	0	0	0	0	415	0.003355243
4	0	0	2383	0	0	2383	2383	0	0	0	0	7149	0	0	0	0	0	98	0	0	0	0	98	0.013708211
	1	0	0	0	2383	2383	0	0	0	0	0	4766	0	0	0	329	0	0	0	0	0	0	329	0.069030634
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#DIV/0!
	3	2383	2383	2383	0	2383	2383	0	0	0	0	11915	127	0	118	0	0	0	0	0	0	0	245	0.020562316
	4	0	2383	0	0	0	2383	0	0	0	0	4766	0	0	0	0	0	153	0	0	0	0	153	0.032102392
	5	2383	0	0	0	0	0	0	0	0	0	2383	448	0	0	0	0	0	0	0	0	0	448	0.187998321
	6	13743	13743	13743	13743	13743	13743	0	13743	13743	13743	123687	244	0	0	116	0	0	0	0	0	0	360	0.002910573
	7	13743	13743	13743	13743	13743	13743	13743	0	13743	13743	123687	163	0	0	169	0	0	0	0	0	0	332	0.002684195
	8	13743	13743	13743	13743	13743	13743	13743	13743	0	13743	123687	235	0	0	96	0	0	0	0	0	0	331	0.002676511
	9	13743	13743	13743	13743	13743	13743	13743	13743	13743	0	123687	171	0	0	112	0	0	0	0	0	0	283	0.002288034
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#DIV/0!
	1	0	0	0	2383	0	0	0	0	0	0	2383	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	2383	0	0	0	0	0	0	2383	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	2383	0	0	0	0	0	2383	0	0	0	0	352	0	0	0	0	0	352	0.147712967
	4	2383	0	0	2383	0	0	0	0	0	0	4766	426	0	0	0	0	0	0	0	0	0	426	0.089383131
	5	13743	13743	13743	13743	13743	0	13743	13743	13743	13743	123687	302	0	0	0	0	0	0	0	0	0	302	0.002441647
	6	13743	13743	13743	13743	13743	13743	0	13743	13743	13743	123687	148	0	162	0	0	0	0	0	0	0	310	0.002506326
	7	13743	13743	13743	13743	13743	13743	13743	0	13743	13743	123687	253	0	90	0	0	0	0	0	0	0	343	0.002775129
	8	13743	13743	13743	13743	13743	13743	13743	13743	0	13743	123687	264	0	91	0	0	0	0	0	0	0	355	0.002870148
	9	13743	13743	13743	13743	13743	13743	13743	13743	13743	0	123687	217	0	87	0	0	0	0	0	23	0	327	0.00264377

Fig 3.3.1: Packet Delivery Ratio under Constant attack

Malicious Nodes	Nodes	0	1	2	3	4	5	6	7	8	9	Total Sent	0	1	2	3	4	5	6	7	8	9	Total Drop	Packet Drop Ratio
0	0	0	2383	2383	2383	0	0	0	0	2383	0	9532	0	132	195	29	0	0	0	0	1544	0	1900	0.199328577
	1	2383	0	2383	0	2383	0	2383	2383	0	0	11915	5	0	194	0	2382	0	79	194	0	0	2854	0.239530004
	2	0	2383	0	0	2383	0	2383	0	0	2383	9532	0	132	0	0	2204	0	79	0	0	150	2565	0.26909358
	3	2383	2383	2383	0	0	0	2383	0	2383	2383	14298	8	133	194	0	0	0	79	0	1544	150	2108	0.147433207
	4	0	0	2383	0	0	0	2383	2383	2383	0	9532	0	0	194	0	0	0	79	194	1544	0	2011	0.210973563
	5	2383	2383	0	0	2383	0	0	0	0	0	7149	4	132	0	0	2382	0	0	0	0	0	2518	0.352217093
	6	2383	2383	2383	2383	2383	2383	0	0	0	0	14298	4	132	194	29	2382	194	0	0	0	0	2935	0.205273465
	7	2383	0	2383	0	2383	0	0	0	0	0	7149	4	0	13	0	2382	0	0	0	0	0	2399	0.335571409
	8	2383	2383	2383	2383	0	0	2383	2383	0	2383	16681	2	132	34	29	0	0	79	194	0	150	620	0.037168035
	9	2383	2383	2383	2383	0	2383	2383	2383	0	0	16681	8	132	194	29	0	194	79	194	0	0	830	0.049757209
1	0	0	2383	0	0	2383	0	0	0	2383	0	7149	0	129	0	0	105	0	0	0	1798	0	2032	0.284235557
	1	2383	0	2383	0	2383	0	0	0	0	0	7149	1	0	62	0	105	0	0	0	0	0	168	0.02349979
	2	0	2383	0	2383	2383	0	0	0	2383	0	9532	0	129	0	1228	106	0	0	0	1798	0	3261	0.342110785
	3	0	2383	0	0	0	0	2383	2383	0	0	7149	0	129	0	0	0	0	78	202	0	0	409	0.057210799
	4	2383	2383	2383	2383	0	0	2383	0	0	0	11915	1	129	81	376	0	0	78	0	0	0	665	0.055812002
	5	0	2383	2383	2383	0	0	2383	0	2383	0	11915	0	129	9	81	0	0	78	0	1798	0	2095	0.175828787
	6	0	2383	2383	0	2383	0	0	2383	2383	0	11915	0	129	13	0	105	0	0	202	1798	0	2247	0.188585816
	7	0	0	0	2383	0	0	2383	0	0	0	4766	0	0	0	25	0	0	78	0	0	0	103	0.021611414
	8	2383	2383	2383	2383	2383	2383	2383	0	0	0	16681	5	129	66	159	105	206	78	0	0	0	748	0.044841436
	9	13743	13743	13743	13743	13743	13743	13743	13743	13743	0	123687	17	769	1007	13742	625	1210	472	1192	10399	0	29433	0.237963569
2	0	0	2383	0	0	0	0	2383	2383	0	0	7149	0	191	0	0	0	0	2378	1681	0	0	4250	0.59448874
	1	0	0	2383	0	2383	2383	0	2383	0	0	9532	0	0	5	0	1546	190	0	1681	0	0	3422	0.359001259
	2	0	0	0	2383	0	0	2383	2383	0	0	7149	0	0	0	2382	0	0	2378	1681	0	0	6441	0.90096517
	3	0	2383	0	0	0	0	2383	2383	0	0	7149	0	190	0	0	0	0	2378	1681	0	0	4249	0.59434886
	4	0	0	0	2383	0	2383	0	0	0	0	4766	0	0	0	2382	0	191	0	0	0	0	2573	0.539865715
	5	2383	2383	0	0	0	0	2383	0	0	0	7149	1	130	0	0	0	0	0	1681	0	0	1812	0.253462023
	6	2383	0	0	0	0	2383	0	2383	0	0	7149	1	0	0	0	0	190	0	1681	0	0	1872	0.261854805
	7	0	2383	2383	2383	0	0	2383	0	0	0	7149	0	0	26	2382	0	0	2378	0	0	0	4786	0.669464261
	8	13743	13743	13743	13743	13743	13743	13743	13743	0	13743	123687	8	1120	1043	9103	8941	1120	13738	9724	0	886	45683	0.369343585
	9	13743	13743	13743	13743	13743	13743	13743	13743	13743	0	123687	17	1120	503	9103	8941	1120	13738	9724	9022	0	53288	0.430829432
3	0	0	2383	2383	0	2383	2383	2383	0	0	0	11915	0	2383	0	0	115	189	202	0	0	0	2889	0.242467478
	1	0	0	0	0	2383	0	0	0	0	0	2383	0	0	0	0	116	0	0	0	0	0	116	0.048678137
	2	2383	0	0	2383	0	0	2383	0	0	0	7149	18	0	0	1504	0	0	202	0	0	0	1724	0.241152609
	3	0	2383	2383	0	2383	0	2383	0	0	0	9532	0	2383	0	0	115	0	202	0	0	0	2700	0.283256399
	4	0	2383	0	0	0	2383	2383	0	0	0	7149	0	2383	0	0	0	189	86	0	0	0	2658	0.371800252
	5	2383	0	2383	0	2383	0	2383	0	0	0	9532	6	0	0	0	115	0	202	0	0	0	323	0.033885858
	6	0	2383	0	2383	0	0	0	0	0	0	4766	0	696	0	1504	0	0	0	0	0	0	2200	0.461603021
	7	13743	13743	13743	13743	13743	13743	13743	0	13743	13743	123687	72	13742	22	8698	688	1120	1170	0	355	940	26807	0.216732559
	8	13743	13743	13743	13743	13743	13743	13743	13743	0	13743	123687	233	13742	22	8698	688	1120	1170	1147	0	940	27760	0.224437491
	9	13743	13743	13743	13743	13743	13743	13743	13743	13743	0	123687	8	13742	22	8698	688	1120	1170	1147	355	0	26950	0.217888703
4	0	0	2383	0	0	2383	2383	0	0	0	0	7149	0	196	0	0	2383	190	0	0	0	0	2769	0.387326899
	1	0	0	0	2383	2383	0	0	0	0	0	4766	0	0	0	21	2383	0	0	0	0	0	2404	0.504406211
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#DIV/0!
	3	2383	2383	2383	0	2383	2383	0	0	0	0	11915	12	195	20	0	2383	190	0	0	0	0	2800	0.234997902
	4	0	2383	0	0	0	2383	0	0	0	0	4766	0	195	0	0	0	190	0	0	0	0	385	0.080780529
	5	2383	0	0	0	0	0	0	0	0	0	2383	4	0	0	0	0	0	0	0	0	0	4	0.001678556
	6	13743	13743	13743	13743	13743	13743	0	13743	13743	13743	123687	17	1147	1727	139	13742	1120	0	1147	9040	1003	29082	0.235125761
	7	13743	13743	13743	13743	13743	13743	13743	0	13743	13743	123687	152	3499	440	139	13742	1120	445	0	9040	1003	29580	0.239152053
	8	13743	13743	13743	13743	13743	13743	13743	13743	0	13743	123687	8	1147	1331	139	13742	1120	445	1147	0	1003	20082	0.162361445
	9	13743	13743	13743	13743	13743	13743	13743	13743	13743	0	123687	81	1147	395	139	13742	1120	445	1147	9040	0	27256	0.22036269
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#DIV/0!
	1	0	0	0	2383	0	0	0	0	0	0	2383	0	0	0	2378	0	0	0	0	0	0	2378	0.997901804
	2	0	0	0	2383	0	0	0	0	0	0	2383	0	0	0	2378	0	0	0	0	0	0	2378	0.997901804
	3	0	0	0	0	2383	0	0	0	0	0	2383	0	0	0	0	108	0	0	0	0	0	108	0.045321024
	4	2383	0	0	2383	0	0	0	0	0	0	4766	24	0	0	2378	0	0	0	0	0	0	2402	0.503986572
	5	13743	13743	13743	13743	13743	0	13743	13743	13743	13743	123687	8	13738	1772	13738	643	0	445	1174	337	10057	41912	0.338855336
	6	13743	13743	13743	13743	13743	13743	0	13743	13743	13743	123687	170	13738	153	13738	643	1111	0	1174	337	10057	41121	0.332460162
	7	13743	13743	13743	13743	13743	13743	13743	0	13743	13743	123687	80	13738	242	13738	643	1111	445	0	337	10057	40391	0.326558167
	8	13743	13743	13743	13743	13743	13743	13743	13743	0	13743	123687	62	13738	233	13738	643	1111	445	1174	0	10057	41201	0.333106955
	9	13743	13743	13743	13743	13743	13743	13743	13743	13743	0	123687	152	13738	278	13738	643	1111	445	1174	337	0	31616	0.255612959

Fig 3.3.2: Packet Drop Ratio under Constant attack

Malicious Nodes	Nodes	0	1	2	3	4	5	6	7	8	9	Total Sent	0	1	2	3	4	5	6	7	8	9	Total Reci	Packet Delivery Ratio
0	0	0	0	2383	2383	2383	0	2383	0	0	0	9532	0	0	212	191	99	0	0	0	0	0	502	0.052664708
	1	0	0	2383	2383	0	2383	2383	2383	2383	2383	16681	0	0	236	216	0	0	0	0	0	67	519	0.031113243
	2	2383	2383	0	2383	0	2383	0	2383	2383	0	14298	225	0	0	206	0	0	0	32	0	0	463	0.032382151
	3	2383	0	2383	0	0	0	0	0	0	2383	9532	184	0	231	0	0	0	0	0	0	64	479	0.050251783
	4	2383	2383	2383	0	0	2383	2383	0	2383	2383	16681	215	0	216	0	0	0	0	0	0	47	478	0.028655356
	5	0	2383	2383	0	2383	0	2383	0	0	0	9532	0	0	240	0	0	0	161	0	0	0	401	0.042068821
	6	2383	2383	2383	0	0	0	0	2383	2383	0	11915	215	0	219	0	0	0	0	0	0	0	434	0.036424675
	7	2383	2383	0	2383	0	2383	0	0	0	0	9532	230	0	0	230	0	0	0	0	0	0	460	0.048258498
	8	0	2383	0	0	0	0	2383	2383	0	2383	9532	0	0	0	0	0	0	253	0	0	163	416	0.043642467
	9	0	2383	0	2383	2383	2383	0	0	2383	0	11915	0	0	0	277	184	0	0	0	62	0	523	0.043894251
1	0	0	0	2383	2383	0	0	0	2383	0	0	7149	0	0	237	155	0	0	0	143	0	0	535	0.074835641
	1	0	0	2383	2383	0	2383	0	0	0	0	7149	0	0	333	238	0	0	0	0	0	0	571	0.079871311
	2	0	2383	0	2383	0	0	0	0	2383	0	7149	0	120	0	47	0	0	0	0	0	0	167	0.02335991
	3	0	0	0	0	2383	0	0	0	0	0	2383	0	0	0	0	128	0	0	0	0	0	128	0.053713806
	4	0	0	0	2383	0	0	0	2383	0	0	4766	0	0	0	357	0	0	0	195	0	0	552	0.115820394
	5	0	0	0	2383	0	0	2383	0	0	0	4766	0	0	0	267	0	0	208	0	0	0	475	0.099664289
	6	2383	0	0	0	0	0	0	0	2383	0	4766	549	0	0	0	0	0	0	0	0	0	549	0.115190936
	7	0	2383	2383	0	2383	2383	2383	0	2383	0	14298	0	96	217	0	0	0	143	0	0	0	456	0.031892572
	8	2383	0	2383	2383	0	0	2383	0	0	0	9532	202	0	167	21	0	0	123	0	0	0	513	0.053818716
	9	8262	8262	8262	8262	8262	8262	8262	8262	8262	0	74358	406	0	0	0	0	0	0	0	0	0	406	0.005460072
2	0	0	2383	2383	2383	2383	0	2383	0	0	0	11915	0	0	0	117	30	0	61	0	0	0	208	0.017456987
	1	0	0	2383	2383	2383	0	2383	2383	0	0	11915	0	0	173	168	0	0	113	0	0	0	454	0.038103231
	2	0	2383	0	0	2383	2383	0	2383	0	0	9532	0	0	0	0	0	67	0	64	0	0	131	0.013743181
	3	0	0	2383	0	2383	2383	0	0	0	0	7149	0	0	291	0	0	74	0	0	0	0	365	0.051056092
	4	2383	2383	0	2383	0	2383	2383	2383	0	0	14298	151	0	0	185	0	0	129	0	0	0	465	0.032522031
	5	2383	2383	0	0	0	0	2383	0	0	0	7149	163	0	0	0	0	0	138	0	0	0	301	0.042103791
	6	0	0	2383	2383	0	0	0	0	0	0	4766	0	0	236	224	0	0	0	0	0	0	460	0.096516995
	7	2383	2383	2383	2383	2383	0	2383	0	0	0	14298	136	0	130	118	32	0	63	0	0	0	479	0.033501189
	8	7569	7569	7569	7569	7569	7569	7569	7569	0	7569	68121	145	0	0	245	0	0	0	0	0	0	390	0.005725107
	9	6894	6894	6894	6894	6894	6894	6894	6894	6894	0	62046	91	0	0	196	0	0	0	0	0	0	287	0.0046256
3	0	0	0	0	2383	2383	2383	2383	0	0	0	9532	0	0	0	182	0	107	0	0	0	0	289	0.030318926
	1	0	0	2383	2383	0	2383	2383	0	0	0	9532	0	0	206	122	0	20	0	0	0	0	348	0.036508603
	2	2383	2383	0	2383	2383	2383	2383	0	0	0	14298	242	92	0	0	0	55	0	0	0	0	389	0.027206602
	3	0	2383	0	0	2383	0	0	0	0	0	4766	0	226	0	0	0	0	0	0	0	0	226	0.047419219
	4	0	2383	2383	2383	0	2383	2383	0	0	0	11915	0	94	242	0	0	57	0	0	0	0	393	0.032983634
	5	0	0	0	2383	0	0	2383	0	0	0	4766	0	0	0	355	0	0	0	0	0	0	355	0.074485942
	6	2383	0	2383	0	0	2383	0	0	0	0	7149	182	0	191	0	0	5	0	0	0	0	378	0.052874528
	7	6885	6885	6885	6885	6885	6885	6885	6885	0	6885	61965	189	0	163	0	0	0	0	0	0	0	352	0.005680626
	8	8262	8262	8262	8262	8262	8262	8262	8262	8262	0	74358	200	0	167	0	0	0	0	0	0	0	367	0.004935582
	9	6885	6885	6885	6885	6885	6885	6885	6885	6885	0	61965	153	0	212	0	0	0	0	0	0	0	365	0.005890422
4	0	0	0	0	0	0	2383	0	0	0	0	2383	0	0	0	0	0	61	0	0	0	0	61	0.025597986
	1	0	0	2383	0	2383	2383	0	0	0	0	7149	0	0	237	0	138	56	0	0	0	0	431	0.060288152
	2	0	0	0	2383	0	2383	0	0	0	0	4766	0	0	0	132	0	0	0	0	0	0	132	0.027696181
	3	2383	2383	2383	0	2383	0	0	0	0	0	9532	66	0	216	0	114	0	0	0	0	0	396	0.041544272
	4	0	2383	2383	0	0	2383	0	0	0	0	7149	0	0	312	0	0	152	0	0	0	0	464	0.064904182
	5	0	0	2383	0	0	0	0	0	0	0	2383	0	0	422	0	0	0	0	0	0	0	422	0.177087705
	6	8262	8262	8262	8262	8262	8262	0	8262	8262	8262	74358	0	0	0	167	0	0	0	0	59	0	226	0.00303935
	7	8262	8262	8262	8262	8262	8262	8262	0	8262	8262	74358	290	0	0	68	0	0	0	0	0	0	358	0.004814546
	8	6894	6894	6894	6894	6894	6894	6894	6894	0	6894	62046	0	0	0	253	0	0	41	0	0	0	294	0.00473842
	9	8262	8262	8262	8262	8262	8262	8262	8262	8262	0	74358	0	0	0	176	0	0	0	0	62	0	238	0.003200732
5	0	0	0	2383	0	2383	0	0	0	0	0	4766	0	0	0	0	206	0	0	0	0	0	206	0.043222828
	1	0	0	0	2383	2383	0	0	0	0	0	4766	0	0	0	169	86	0	0	0	0	0	255	0.053503987
	2	0	2383	0	0	0	0	0	0	0	0	2383	0	309	0	0	0	0	0	0	0	0	309	0.129668485
	3	0	0	2383	0	0	0	0	0	0	0	2383	0	0	0	0	0	0	0	0	0	0	0	0
	4	2383	0	0	0	0	0	0	0	0	0	2383	462	0	0	0	0	0	0	0	0	0	462	0.193873269
	5	8262	8262	8262	8262	8262	0	8262	8262	8262	8262	74358	253	0	0	0	0	0	0	0	6	0	259	0.003483149
	6	8262	8262	8262	8262	8262	8262	0	8262	8262	8262	74358	289	0	0	0	0	0	0	0	0	0	289	0.003886603
	7	6894	6894	6894	6894	6894	6894	6894	0	6894	6894	62046	298	0	0	0	0	0	0	0	0	0	298	0.004802888
	8	6894	6894	6894	6894	6894	6894	6894	6894	0	6894	62046	315	0	0	0	0	0	0	0	0	0	315	0.005076878
	9	6885	6885	6885	6885	6885	6885	6885	6885	6885	0	61965	324	0	0	0	0	0	0	0	24	0	348	0.005616074

Fig 3.3.3: Packet Delivery Ratio under Periodic attack

Malicious Nodes	Nodes	0	1	2	3	4	5	6	7	8	9	Total Sent	0	1	2	3	4	5	6	7	8	9	Total Drop	Packet Drop Ratio
0	0	0	0	2383	2383	2383	0	2383	0	0	0	9532	0	0	0	20	112	0	2382	0	0	0	2514	0.263743181
	1	0	0	2383	2383	0	2383	2383	2383	2383	2383	16681	0	0	0	20	0	1834	2383	1578	2383	168	8366	0.501528685
	2	2383	2383	0	2383	0	2383	0	2383	2383	0	14298	2	168	0	20	0	1834	0	194	2382	0	4600	0.321723318
	3	2383	0	2383	0	0	0	0	0	2383	2383	9532	48	0	1	0	0	0	0	0	1903	168	2120	0.222408728
	4	2383	2383	2383	0	0	2383	2383	0	2383	2383	16681	1	1	613	0	0	0	1834	2383	0	2383	168	7382	0.442539416
	5	0	2383	2383	0	2383	0	2383	0	0	0	9532	0	168	0	0	444	0	78	0	0	0	690	0.072387747
	6	2383	2383	2383	0	0	0	0	2383	2383	0	11915	5	168	0	0	0	0	0	1578	2382	0	4133	0.346873689
	7	2383	2383	0	2383	0	2383	0	0	0	0	9532	20	168	0	20	0	1834	0	0	0	0	2042	0.214225766
	8	0	2383	0	0	0	0	2383	2383	0	2383	9532	0	168	0	0	0	0	78	1578	0	168	1992	0.208980277
	9	0	2383	0	2383	2383	2383	0	0	2383	0	11915	0	629	0	20	112	1834	0	0	234	0	2829	0.237431809
1	0	0	0	2383	2383	0	0	0	2383	0	0	7149	0	0	97	179	0	0	0	191	0	0	467	0.065323822
	1	0	0	2383	2383	0	2383	0	0	0	0	7149	0	0	8	102	0	1606	0	0	0	0	1716	0.240033571
	2	0	2383	0	2383	0	0	0	0	2383	0	7149	0	130	0	203	0	0	0	0	1437	0	1770	0.247587075
	3	0	0	0	0	2383	0	0	0	0	0	2383	0	0	0	0	1705	0	0	0	0	0	1705	0.715484683
	4	0	0	0	2383	0	0	0	2383	0	0	4766	0	0	0	30	0	0	0	191	0	0	221	0.046370122
	5	0	0	0	2383	0	0	2383	0	0	0	4766	0	0	0	24	0	0	82	0	0	0	106	0.022240873
	6	2383	0	0	0	0	0	0	0	2383	0	4766	1	0	0	0	0	0	0	0	1437	0	1438	0.30172052
	7	0	2383	2383	0	2383	2383	2383	0	2383	0	14298	0	130	8	0	1705	1606	82	0	1437	0	4968	0.347461183
	8	2383	0	2383	2383	0	0	2383	0	0	0	9532	4	0	38	184	0	0	82	0	0	0	308	0.032312211
	9	8262	8262	8262	8262	8262	8262	8262	8262	8262	0	74358	17	778	566	2754	5503	1111	499	1129	5503	0	17860	0.240189354
2	0	0	2383	2383	2383	2383	0	2383	0	0	0	11915	0	152	273	18	104	0	73	0	0	0	620	0.05203525
	1	0	0	2383	2383	2383	0	2383	2383	0	0	11915	0	0	13	18	244	0	73	244	0	0	592	0.049685271
	2	0	2383	0	0	2383	2383	0	2383	0	0	9532	0	152	0	0	244	242	0	244	0	0	882	0.092530424
	3	0	0	2383	0	2383	2383	0	0	0	0	7149	0	0	25	0	244	242	0	0	0	0	511	0.071478528
	4	2383	2383	0	2383	0	2383	2383	2383	0	0	14298	52	152	0	18	0	242	73	244	0	0	781	0.054623024
	5	2383	2383	0	0	0	0	2383	0	0	0	7149	49	152	0	0	0	0	73	0	0	0	274	0.038327039
	6	0	0	2383	2383	0	0	0	0	0	0	4766	0	0	6	18	0	0	0	0	0	0	24	0.005035669
	7	2383	2383	2383	2383	2383	0	2383	0	0	0	14298	1	152	7	18	104	0	73	0	0	0	355	0.024828647
	8	7569	7569	7569	7569	7569	7569	7569	7569	0	7569	68121	233	904	503	130	625	1426	445	1435	0	904	6605	0.096959821
	9	6894	6894	6894	6894	6894	6894	6894	6894	6894	0	62046	242	787	485	130	1372	1372	445	1372	4477	0	10682	0.172162589
3	0	0	0	0	2383	2383	2383	2383	0	0	0	9532	0	0	0	114	1550	188	1736	0	0	0	3588	0.376416282
	1	0	0	2383	2383	0	2383	2383	0	0	0	9532	0	0	3	87	0	188	1736	0	0	0	2014	0.211288292
	2	2383	2383	0	2383	2383	2383	2383	0	0	0	14298	2	152	0	747	1550	188	1736	0	0	0	4375	0.305986851
	3	0	2383	0	0	2383	0	0	0	0	0	4766	0	152	0	0	1550	0	0	0	0	0	1702	0.357112883
	4	0	2383	2383	2383	0	2383	2383	0	0	0	11915	0	152	3	533	0	188	1736	0	0	0	2612	0.219219471
	5	0	0	0	2383	0	0	2383	0	0	0	4766	0	0	0	26	0	0	1736	0	0	0	1762	0.369702056
	6	2383	0	2383	0	0	2383	0	0	0	0	7149	12	0	3	0	0	188	0	0	0	0	203	0.02839558
	7	6885	6885	6885	6885	6885	6885	6885	0	6885	6885	61965	18	904	40	6884	5539	1111	6610	0	6871	6745	34722	0.560348584
	8	8262	8262	8262	8262	8262	8262	8262	8262	0	8262	74358	9	904	40	8261	5503	1111	5503	1156	0	5503	27990	0.376422174
	9	6885	6885	6885	6885	6885	6885	6885	6885	6885	0	61965	108	904	40	6884	5539	1111	6610	1156	6871	0	29223	0.471604938
4	0	0	0	0	0	0	2383	0	0	0	0	2383	0	0	0	0	0	189	0	0	0	0	189	0.079311792
	1	0	0	2383	0	2383	2383	0	0	0	0	7149	0	0	8	0	107	189	0	0	0	0	304	0.04252343
	2	0	0	0	2383	0	2383	0	0	0	0	4766	0	0	0	35	0	189	0	0	0	0	224	0.04699958
	3	2383	2383	2383	0	2383	0	0	0	0	0	9532	156	1483	6	0	107	0	0	0	0	0	1752	0.18380193
	4	0	2383	2383	0	0	2383	0	0	0	0	7149	0	1483	30	0	0	189	0	0	0	0	1702	0.238075255
	5	0	0	2383	0	0	0	0	0	0	0	2383	0	0	6	0	0	0	0	0	0	0	6	0.002517835
	6	8262	8262	8262	8262	8262	8262	0	8262	8262	8262	74358	620	5503	1170	229	643	1111	0	1129	337	895	11637	0.156499637
	7	8262	8262	8262	8262	8262	8262	8262	0	8262	8262	74358	8	5503	2754	229	643	1111	436	0	337	895	11916	0.160251755
	8	6894	6894	6894	6894	6894	6894	6894	6894	0	6894	62046	521	4468	1259	229	643	1111	436	1129	0	895	10691	0.172307643
	9	8262	8262	8262	8262	8262	8262	8262	8262	8262	0	74358	557	5503	2222	229	643	1111	436	1129	337	0	12167	0.163627316
5	0	0	0	2383	0	2383	0	0	0	0	0	4766	0	0	2382	0	112	0	0	0	0	0	2494	0.523289971
	1	0	0	0	2383	2383	0	0	0	0	0	4766	0	0	0	29	112	0	0	0	0	0	141	0.029584557
	2	0	2383	0	0	0	0	0	0	0	0	2383	0	149	0	0	0	0	0	0	0	0	149	0.062526227
	3	0	0	2383	0	0	0	0	0	0	0	2383	0	0	2382	0	0	0	0	0	0	0	2382	0.999580361
	4	2383	0	0	0	0	0	0	0	0	0	2383	12	0	0	0	0	0	0	0	0	0	12	0.005035669
	5	8262	8262	8262	8262	8262	8262	0	8262	8262	8262	74358	125	886	8261	1340	670	0	490	1291	364	958	14385	0.193455983
	6	8262	8262	8262	8262	8262	8262	8262	0	8262	8262	74358	53	886	8261	2754	670	8257	0	1291	364	958	23494	0.315957933
	7	6894	6894	6894	6894	6894	6894	6894	6894	0	6894	62046	8	886	6894	2754	670	6889	490	0	364	958	19913	0.320933303
	8	6894	6894	6894	6894	6894	6894	6894	6894	6894	0	62046	26	886	6894	5966	670	6889	490	5503	0	958	28282	0.455823099
	9	6885	6885	6885	6885	6885	6885	6885	6885	6885	0	61965	72	886	6885	2195	670	6880	490	6637	364	0	25079	0.404728476

Fig 3.3.4: Packet Drop Ratio under Periodic attack

Malicious Nodes	Nodes	0	1	2	3	4	5	6	7	8	9	Total Sent	0	1	2	3	4	5	6	7	8	9	Total Reci	Packet Delivery Ratio
0	0	0	0	0	2383	2383	0	2383	0	0	2383	9532	0	0	0	229	0	0	261	0	0	0	490	0.051405791
	1	2383	0	0	0	0	2383	2383	0	2383	2383	11915	259	0	0	0	0	0	0	0	221	0	480	0.040285355
	2	0	0	0	2383	0	2383	0	2383	2383	0	9532	0	0	0	230	0	0	0	57	209	0	496	0.05203525
	3	0	0	0	0	2383	2383	0	0	2383	2383	9532	0	0	0	0	0	0	0	0	339	0	339	0.035564415
	4	2383	2383	2383	2383	0	0	2383	2383	2383	0	16681	178	50	177	0	0	0	0	0	127	0	532	0.031892572
	5	2383	2383	0	2383	2383	0	0	0	0	0	9532	225	118	0	222	0	0	0	0	0	0	565	0.059274024
	6	0	2383	0	0	0	2383	0	0	0	0	4766	0	422	0	0	0	0	0	0	0	0	422	0.088543852
	7	0	2383	0	0	2383	2383	2383	0	0	2383	11915	0	277	0	0	0	0	330	0	0	0	607	0.050944188
	8	2383	0	0	2383	0	0	2383	2383	0	2383	11915	272	0	0	175	0	0	0	66	0	0	513	0.043054973
	9	2383	2383	2383	2383	0	0	0	0	0	0	9532	225	98	225	0	0	0	0	0	0	0	548	0.057490558
1	0	0	2383	2383	0	0	0	0	2383	2383	0	9532	0	263	0	0	0	0	0	235	0	0	498	0.052245069
	1	0	0	2383	2383	2383	2383	2383	0	0	0	11915	0	0	0	194	72	0	140	0	0	0	406	0.034074696
	2	2383	0	0	0	0	0	2383	0	2383	0	7149	321	0	0	0	0	0	246	0	0	0	567	0.079311792
	3	0	0	2383	0	2383	0	0	2383	2383	0	9532	0	0	0	0	132	0	0	71	223	0	426	0.044691565
	4	2383	0	2383	0	0	2383	0	2383	0	0	9532	403	0	0	0	0	0	0	194	0	0	597	0.062631137
	5	2383	2383	2383	2383	0	0	0	0	0	0	9532	287	107	0	0	0	0	0	0	0	0	394	0.041334452
	6	2383	2383	2383	0	0	0	0	2383	2383	0	11915	302	122	0	0	0	0	0	92	0	0	516	0.043306756
	7	0	2383	2383	2383	2383	0	2383	0	0	0	11915	0	20	0	172	51	0	118	0	0	0	361	0.030297944
	8	2383	0	2383	2383	2383	2383	0	2383	0	0	14298	224	0	0	167	79	0	0	18	0	0	488	0.034130648
	9	8946	8946	8946	8946	8946	8946	8946	8946	8946	0	80514	415	0	0	0	0	0	0	0	0	0	415	0.005154383
2	0	0	0	0	0	0	0	0	2383	0	0	2383	0	0	0	0	0	0	0	262	0	0	262	0.109945447
	1	0	0	2383	2383	2383	2383	2383	0	0	0	11915	0	0	215	202	0	28	0	0	0	0	445	0.037347881
	2	0	2383	0	2383	2383	0	2383	0	0	0	9532	0	85	0	196	0	0	0	0	0	0	281	0.029479648
	3	2383	0	0	0	0	2383	0	2383	0	0	7149	206	0	0	0	0	68	0	64	0	0	338	0.04727934
	4	2383	2383	0	0	0	0	0	0	0	0	4766	308	180	0	0	0	0	0	0	0	0	488	0.102391943
	5	2383	2383	2383	0	2383	0	2383	0	0	0	11915	230	102	190	0	0	0	0	0	0	0	522	0.043810323
	6	0	2383	0	0	2383	2383	0	2383	0	0	9532	0	190	0	0	0	128	0	125	0	0	443	0.046475031
	7	2383	0	2383	2383	2383	0	0	0	0	0	9532	92	0	126	129	0	0	0	0	0	0	347	0.036403693
	8	8946	8946	8946	8946	8946	8946	8946	8946	0	8946	80514	117	0	99	131	0	0	0	0	0	0	347	0.004309809
	9	8937	8937	8937	8937	8937	8937	8937	8937	8937	0	80433	154	0	145	158	0	0	0	0	0	0	457	0.005681748
3	0	0	2383	0	0	2383	0	0	0	0	0	4766	0	145	0	0	173	0	0	0	0	0	318	0.066722619
	1	2383	0	0	0	2383	2383	2383	0	0	0	9532	153	0	0	0	57	0	61	0	0	0	271	0.02843055
	2	2383	2383	0	0	2383	0	2383	0	0	0	9532	150	23	0	0	53	0	56	0	0	0	282	0.029584557
	3	2383	0	0	0	2383	0	0	0	0	0	4766	225	0	0	0	125	0	0	0	0	0	350	0.073436844
	4	2383	2383	2383	2383	0	2383	2383	0	0	0	14298	104	23	151	0	0	0	55	0	0	0	333	0.023289971
	5	0	2383	2383	2383	2383	0	0	0	0	0	9532	0	72	201	19	101	0	0	0	0	0	393	0.041229543
	6	2383	2383	2383	0	0	2383	0	0	0	0	9532	169	40	168	0	0	0	0	0	0	0	377	0.039550986
	7	6192	6192	6192	6192	6192	6192	6192	0	6192	6192	55728	208	0	194	0	0	0	0	0	0	0	402	0.007213609
	8	7569	7569	7569	7569	7569	7569	7569	7569	0	7569	68121	199	0	182	0	0	0	0	0	0	0	381	0.005592989
	9	8253	8253	8253	8253	8253	8253	8253	8253	8253	0	74277	208	0	200	0	0	0	0	0	0	0	408	0.005492952
4	0	0	0	2383	0	2383	0	0	0	0	0	4766	0	0	219	0	0	0	0	0	0	0	219	0.045953083
	1	0	0	2383	2383	2383	2383	0	0	0	0	9532	0	0	271	0	0	78	0	0	0	0	349	0.036613512
	2	2383	0	0	2383	2383	0	0	0	0	0	7149	193	0	0	148	0	0	0	0	0	0	341	0.047698979
	3	0	0	0	0	0	2383	0	0	0	0	2383	0	0	0	0	0	122	0	0	0	0	122	0.051195971
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#DIV/0!
	5	0	0	0	2383	2383	0	0	0	0	0	4766	0	0	0	0	370	0	0	0	0	0	370	0.077633235
	6	6876	6876	6876	6876	6876	6876	0	6876	6876	6876	61884	172	0	169	0	0	0	0	0	0	0	341	0.00551031
	7	4815	4815	4815	4815	4815	4815	4815	0	4815	4815	43335	90	0	248	0	0	0	0	0	0	0	338	0.0077997
	8	9630	9630	9630	9630	9630	9630	9630	9630	0	9630	86670	232	0	212	0	0	0	0	0	0	0	444	0.00512288
	9	6201	6201	6201	6201	6201	6201	6201	6201	6201	0	55809	134	0	212	0	0	0	0	0	0	0	346	0.006199717
5	0	0	2383	0	2383	0	0	0	0	0	0	4766	0	0	0	72	0	0	0	0	0	0	72	0.015107008
	1	2383	0	0	2383	2383	0	0	0	0	0	7149	167	0	0	151	0	0	0	0	0	0	318	0.044481746
	2	0	0	0	2383	2383	0	0	0	0	0	4766	0	0	0	237	0	0	0	0	0	0	237	0.049722735
	3	2383	0	0	0	2383	0	0	0	0	0	4766	338	0	0	0	0	0	0	0	0	0	338	0.07091901
	4	2383	0	0	2383	0	0	0	0	0	0	4766	158	0	0	146	0	0	0	0	0	0	304	0.063785145
	5	8271	8271	8271	8271	8271	0	8271	8271	8271	8271	74439	14	0	0	129	0	0	0	0	0	0	143	0.001921036
	6	7569	7569	7569	7569	7569	7569	0	7569	7569	7569	68121	109	0	0	195	0	0	0	0	0	0	304	0.004462647
	7	7569	7569	7569	7569	7569	7569	7569	0	7569	7569	68121	217	0	0	170	0	0	0	0	0	0	387	0.005681068
	8	6201	6201	6201	6201	6201	6201	6201	6201	0	6201	55809	232	0	0	113	0	0	0	0	0	0	345	0.006181799
	9	7569	7569	7569	7569	7569	7569	7569	7569	7569	0	68121	109	0	0	203	0	0	0	0	0	0	312	0.004580085

Fig 3.3.5: Packet Delivery Ratio under Random attack

Malicious Nodes	Nodes	0	1	2	3	4	5	6	7	8	9	Total Sent	0	1	2	3	4	5	6	7	8	9	Total Drop	Packet Drop Ratio
0	0	0	0	0	2383	2383	0	2383	0	0	2383	9532	0	0	0	313	1661	0	280	0	0	1953	4207	0.441355434
	1	2383	0	0	0	0	2383	2383	0	2383	2383	11915	18	0	0	0	0	1737	1778	0	55	1953	5541	0.465044062
	2	0	0	0	2383	0	2383	0	2383	2383	0	9532	0	0	0	34	0	1737	0	206	54	0	2031	0.213071758
	3	0	0	0	0	2383	2383	0	0	2383	2383	9532	0	0	0	0	1661	1737	0	0	54	1953	5405	0.567037348
	4	2383	2383	2383	2383	0	0	2383	2383	2383	0	16681	4	131	4	325	0	0	2382	206	54	0	3106	0.186199868
	5	2383	2383	0	2383	2383	0	0	0	0	0	9532	24	131	0	27	1661	0	0	0	0	0	1843	0.19334872
	6	0	2383	0	0	0	2383	0	0	0	0	4766	0	131	0	0	0	1737	0	0	0	0	1868	0.391942929
	7	0	2383	0	0	2383	2383	2383	0	0	2383	11915	0	131	0	0	1661	1737	78	0	0	1953	5560	0.466638691
	8	2383	0	0	2383	0	0	2383	2383	0	2383	11915	1	0	0	97	0	0	512	206	0	1953	2769	0.232396139
	9	2383	2383	2383	2383	0	0	0	0	0	0	9532	4	131	4	323	0	0	0	0	0	0	462	0.048468317
1	0	0	2383	2383	0	0	0	0	2383	2383	0	9532	0	182	1638	0	0	0	0	210	2382	0	4412	0.462861939
	1	0	0	2383	2383	2383	2383	2383	0	0	0	11915	0	0	1638	28	149	2378	81	0	0	0	4274	0.358707512
	2	2383	0	0	0	0	0	2383	0	2383	0	7149	6	0	0	0	0	0	81	0	2382	0	2469	0.345362988
	3	0	0	2383	0	2383	0	0	2383	2383	0	9532	0	0	1638	0	149	0	0	210	58	0	2055	0.215589593
	4	2383	0	2383	0	0	2383	0	2383	0	0	9532	2	0	1638	0	0	2378	0	210	0	0	4228	0.44355854
	5	2383	2383	2383	2383	0	0	0	0	0	0	9532	1	181	1638	518	0	0	0	0	0	0	2338	0.24527906
	6	2383	2383	2383	0	0	0	0	2383	2383	0	11915	1	181	1638	0	0	0	0	210	2382	0	4412	0.370289551
	7	0	2383	2383	2383	2383	0	2383	0	0	0	11915	0	181	1638	29	149	0	82	0	0	0	2079	0.174485942
	8	2383	0	2383	2383	2383	2383	0	2383	0	0	14298	4	0	1638	61	149	2378	0	210	0	0	4440	0.310532942
	9	8946	8946	8946	8946	8946	8946	8946	8946	8946	0	80514	36	1075	6871	8946	886	8941	499	1237	8945	0	37436	0.464962615
2	0	0	0	0	0	0	0	0	2383	0	0	2383	0	0	0	0	0	0	0	194	0	0	194	0.081409987
	1	0	0	2383	2383	2383	2383	2383	0	0	0	11915	0	0	5	18	1509	191	2382	0	0	0	4105	0.34452371
	2	0	2383	0	2383	2383	0	2383	0	0	0	9532	0	129	0	18	1509	0	2382	0	0	0	4038	0.423625682
	3	2383	0	0	0	0	2383	0	2383	0	0	7149	53	0	0	0	0	191	0	194	0	0	438	0.06126731
	4	2383	2383	0	0	0	0	0	0	0	0	4766	1	129	0	0	0	0	0	0	0	0	130	0.027276542
	5	2383	2383	2383	0	2383	0	2383	0	0	0	11915	1	129	40	0	1509	0	2382	0	0	0	4061	0.340830885
	6	0	2383	0	0	2383	2383	0	2383	0	0	9532	0	129	0	0	1509	191	0	194	0	0	2023	0.21223248
	7	2383	0	2383	2383	2383	0	0	0	0	0	9532	56	0	21	18	1509	0	0	0	0	0	1604	0.168275283
	8	8946	8946	8946	8946	8946	8946	8946	8946	0	8946	80514	145	769	162	130	6187	1129	8945	1147	0	6187	24801	0.308033385
	9	8937	8937	8937	8937	8937	8937	8937	8937	8937	0	80433	134	769	143	130	5494	1129	8936	1147	355	0	18237	0.226735295
3	0	0	2383	0	0	2383	0	0	0	0	0	4766	0	130	0	0	101	0	0	0	0	0	231	0.048468317
	1	2383	0	0	0	2383	2383	2383	0	0	0	9532	5	0	0	0	101	2382	97	0	0	0	2585	0.271191775
	2	2383	2383	0	0	2383	0	2383	0	0	0	9532	4	131	0	0	101	0	97	0	0	0	333	0.034934956
	3	2383	0	0	0	2383	0	0	0	0	0	4766	2	0	0	0	101	0	0	0	0	0	103	0.021611414
	4	2383	2383	2383	2383	0	2383	2383	0	0	0	14298	49	130	2	183	0	2382	97	0	0	0	2843	0.198838998
	5	0	2383	2383	2383	2383	0	0	0	0	0	9532	0	130	1	183	101	0	0	0	0	0	415	0.043537558
	6	2383	2383	2383	0	0	2383	0	0	0	0	9532	1	130	1	0	0	2382	0	0	0	0	2514	0.263743181
	7	6192	6192	6192	6192	6192	6192	6192	0	6192	6192	55728	26	778	31	1084	607	6191	589	0	373	6187	15866	0.284704278
	8	7569	7569	7569	7569	7569	7569	7569	7569	0	7569	68121	17	778	31	1084	607	7568	589	1156	0	6880	18710	0.274658329
	9	8253	8253	8253	8253	8253	8253	8253	8253	8253	0	74277	26	778	31	1084	607	8252	589	1156	373	0	12896	0.173620367
4	0	0	0	2383	0	2383	0	0	0	0	0	4766	0	0	1	0	1291	0	0	0	0	0	1292	0.271086865
	1	0	0	2383	2383	2383	2383	0	0	0	0	9532	0	0	0	1512	664	192	0	0	0	0	2368	0.248426353
	2	2383	0	0	2383	2383	0	0	0	0	0	7149	5	0	0	50	976	0	0	0	0	0	1031	0.144215974
	3	0	0	0	0	0	2383	0	0	0	0	2383	0	0	0	0	0	195	0	0	0	0	195	0.081829627
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	#DIV/0!
	5	0	0	0	2383	2383	0	0	0	0	0	4766	0	0	0	1512	115	0	0	0	0	0	1627	0.341376416
	6	6876	6876	6876	6876	6876	6876	0	6876	6876	6876	61884	26	2753	22	4810	6875	1138	0	6871	337	904	23736	0.383556331
	7	4815	4815	4815	4815	4815	4815	4815	0	4815	4815	43335	180	1376	22	2749	4814	1138	535	0	337	904	12055	0.278181608
	8	9630	9630	9630	9630	9630	9630	9630	9630	0	9630	86670	8	2312	22	6007	9629	1138	904	9625	0	904	30549	0.352474905
	9	6201	6201	6201	6201	6201	6201	6201	6201	6201	0	55809	108	3437	22	4810	6201	1138	535	6187	337	0	22775	0.408088301
5	0	0	2383	0	2383	0	0	0	0	0	0	4766	0	138	0	18	0	0	0	0	0	0	156	0.032731851
	1	2383	0	0	2383	2383	0	0	0	0	0	7149	5	0	0	20	1685	0	0	0	0	0	1710	0.239194293
	2	0	0	0	2383	2383	0	0	0	0	0	4766	0	0	0	19	1685	0	0	0	0	0	1704	0.357532522
	3	2383	0	0	0	2383	0	0	0	0	0	4766	5	0	0	0	1685	0	0	0	0	0	1690	0.354595048
	4	2383	0	0	2383	0	0	0	0	0	0	4766	6	0	0	18	0	0	0	0	0	0	24	0.005035669
	5	8271	8271	8271	8271	8271	0	8271	8271	8271	8271	74439	252	823	8270	132	7006	0	463	6367	6367	886	30566	0.41061809
	6	7569	7569	7569	7569	7569	7569	0	7569	7569	7569	68121	224	823	7568	130	5494	1138	0	5494	5494	886	27251	0.400038167
	7	7569	7569	7569	7569	7569	7569	7569	0	7569	7569	68121	89	823	7568	130	4945	1138	463	0	337	886	16379	0.240439806
	8	6201	6201	6201	6201	6201	6201	6201	6201	0	6201	55809	17	823	6200	130	4945	1138	463	1156	0	886	15758	0.282355892
	9	7569	7569	7569	7569	7569	7569	7569	7569	7569	0	68121	233	823	7568	130	4261	1138	463	4126	4126	0	22868	0.335696775

Fig 3.3.6: Packet Drop Ratio under Random attack

3.4 APPLYING PYTHON SCRIPTS

By using these excel sheets we will be plotting the packet delivery ratios and packet drop ratios for all the nodes in order to detect the malicious node from normal nodes using these below python scripts.

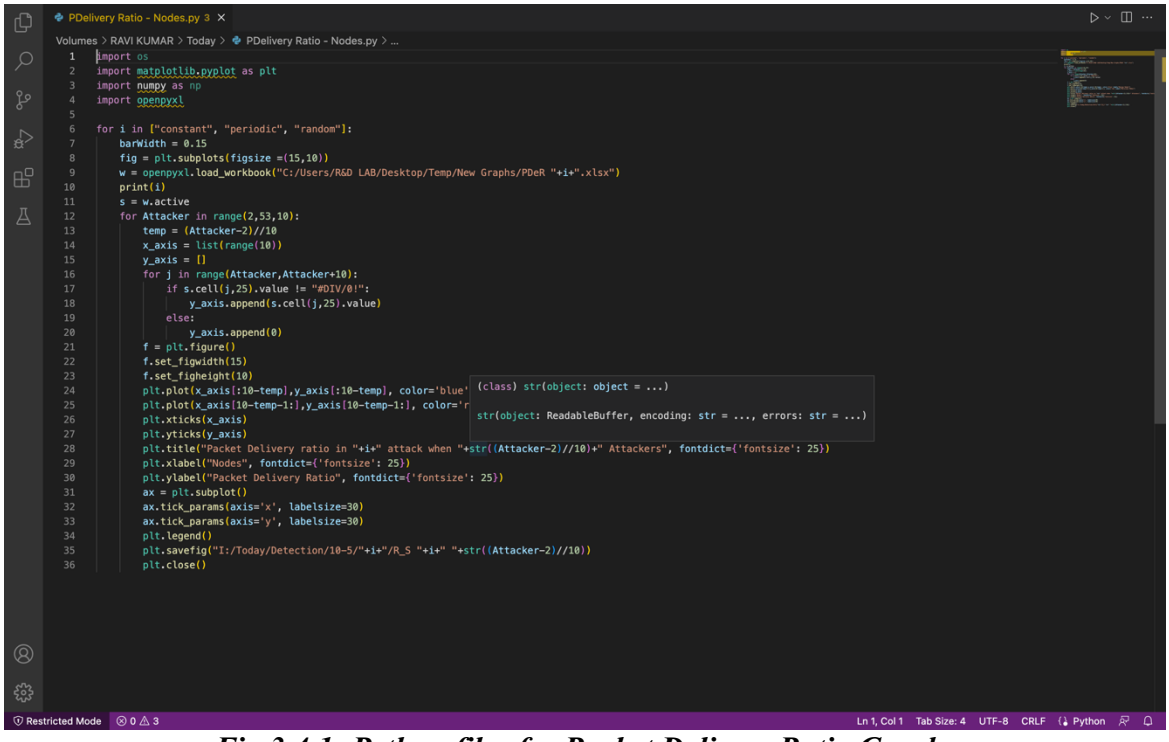

Fig 3.4.1: Python files for Packet Delivery Ratio Graphs

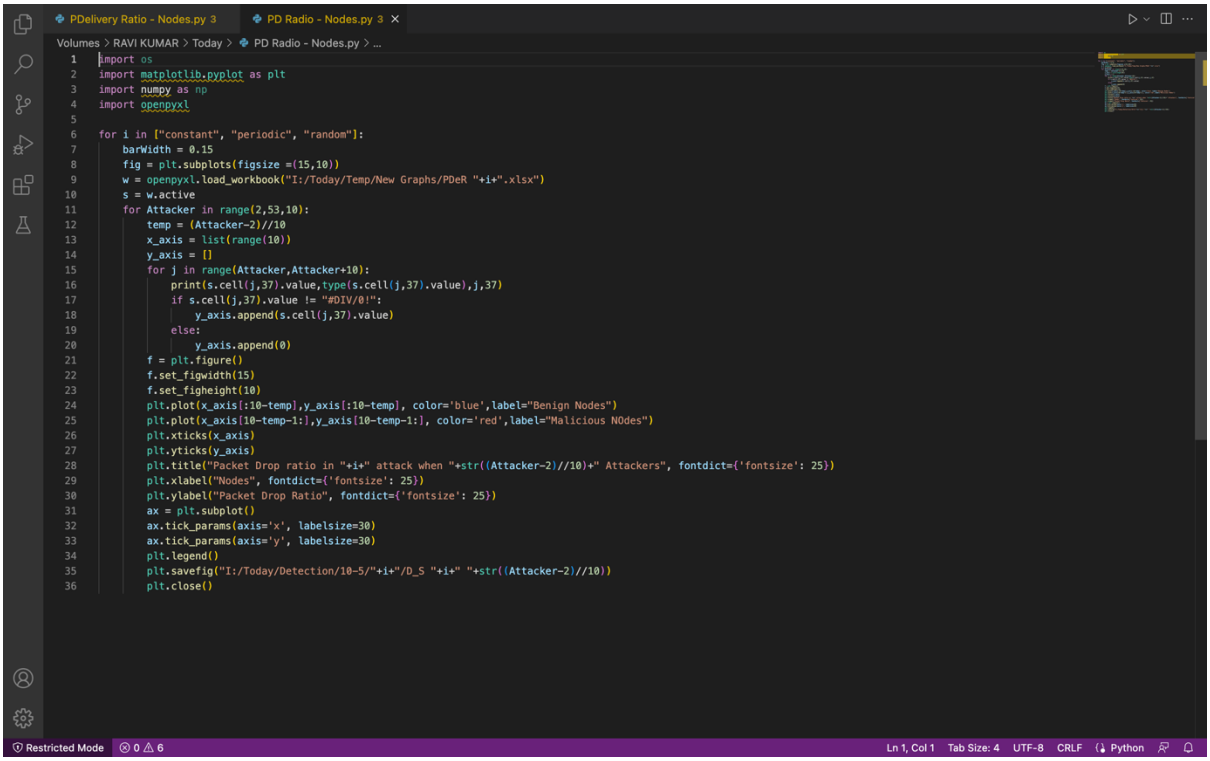

Fig 3.4.2: Python files for Packet Drop Ratio Graphs

CHAPTER – 4

RESULTS AND DISCUSSION

4.1 *Results of Packets Sent, Packet Received and Packets Dropped*

Using the trace files generated, the following graphs which are plotted between number of packets sent and simulation time, number of packets received and simulation time and number of packets dropped and simulation time for every node including malicious nodes. This helps us in differentiating characters among all the nodes.

Constant Attack:

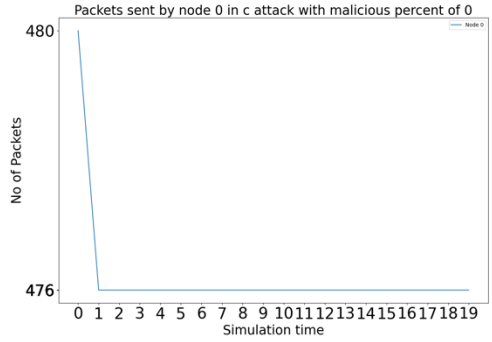

Fig 4.1.0.C.S.0: Packets Sent by Node 0 with 0 attackers in Constant attack.

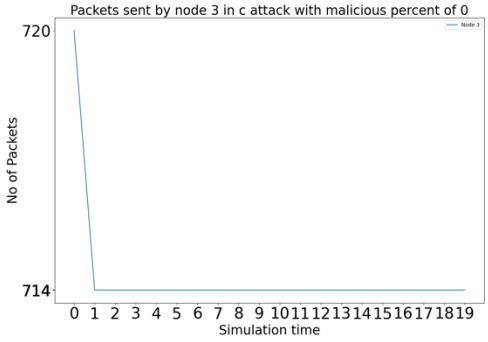

Fig 4.1.0.C.S.3: Packets Sent by Node 3 with 0 attackers in Constant attack.

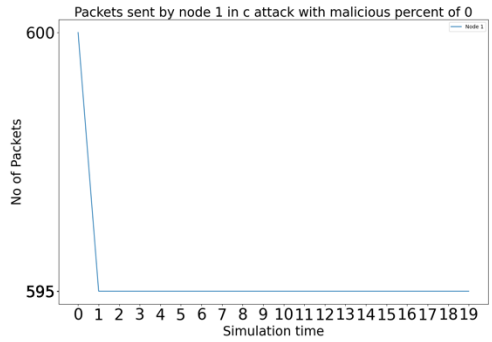

Fig 4.1.0.C.S.1: Packets Sent by Node 1 with 0 attackers in Constant attack.

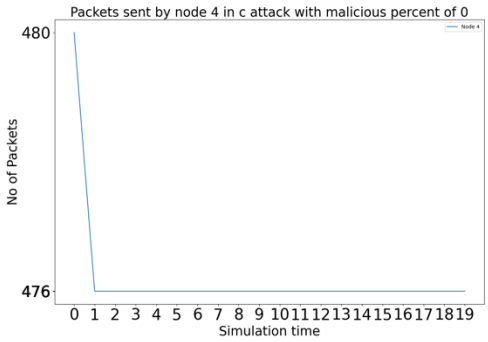

Fig 4.1.0.C.S.4: Packets Sent by Node 4 with 0 attackers in Constant attack.

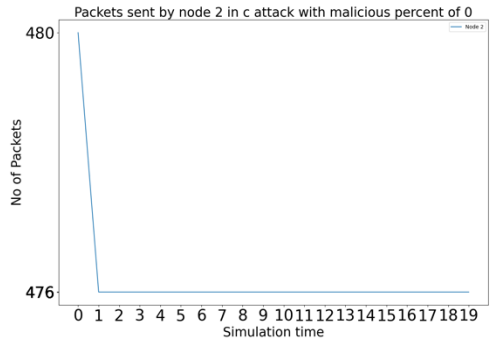

Fig 4.1.0.C.S.2: Packets Sent by Node 2 with 0 attackers in Constant attack.

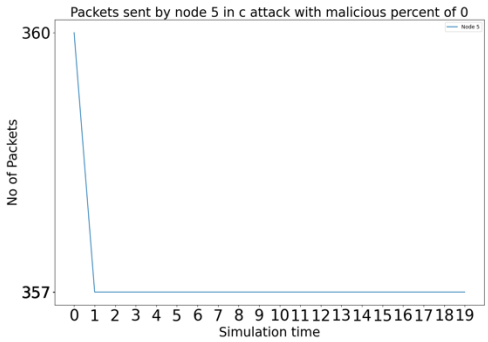

Fig 4.1.0.C.S.5: Packets Sent by Node 5 with 0 attackers in Constant attack.

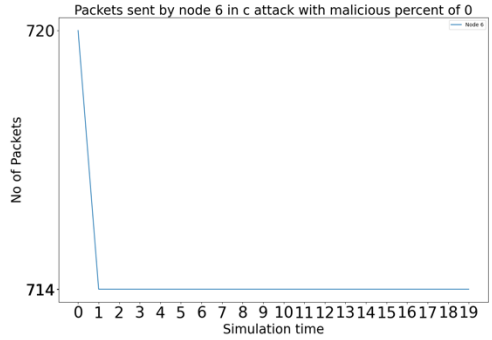

Fig 4.1.0.C.S.6: Packets Sent by Node 6 with 0 attackers in Constant attack.

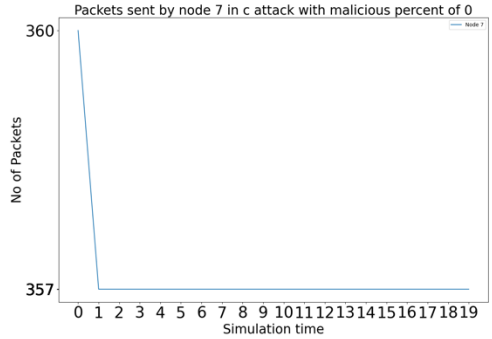

Fig 4.1.0.C.S.7: Packets Sent by Node 7 with 0 attackers in Constant attack.

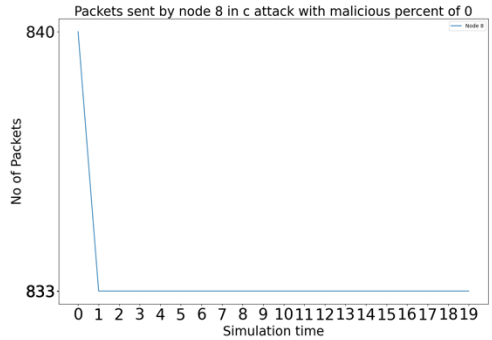

Fig 4.1.0.C.S.8: Packets Sent by Node 8 with 0 attackers in Constant attack.

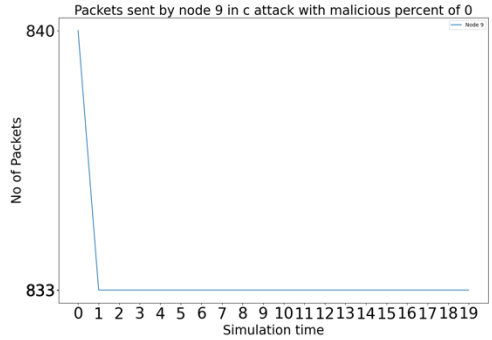

Fig 4.1.0.C.S.9: Packets Sent by Node 9 with 0 attackers in Constant attack.

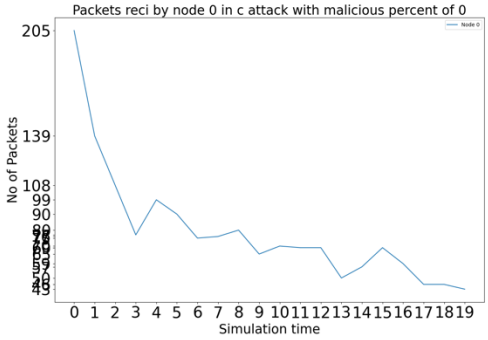

Fig 4.1.0.C.R.0: Packets Received by Node 0 with 0 attackers in Constant attack.

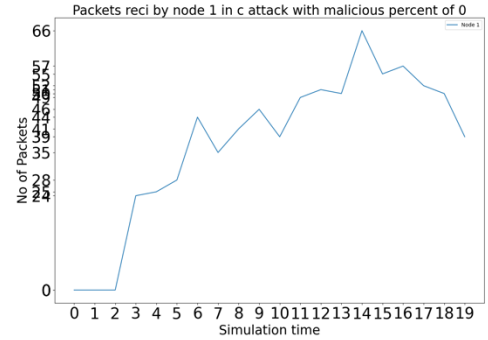

Fig 4.1.0.C.R.1: Packets Received by Node 1 with 0 attackers in Constant attack.

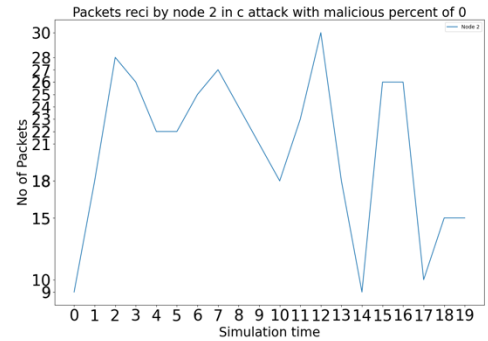

Fig 4.1.0.C.R.2: Packets Received by Node 2 with 0 attackers in Constant attack.

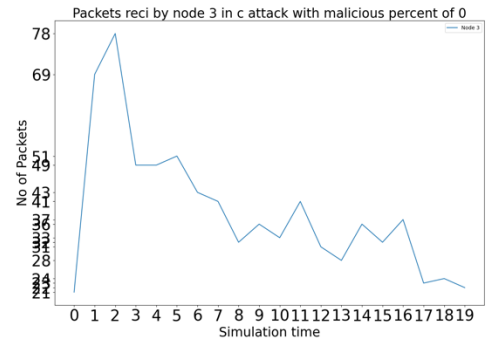

Fig 4.1.0.C.R.3: Packets Received by Node 3 with 0 attackers in Constant attack.

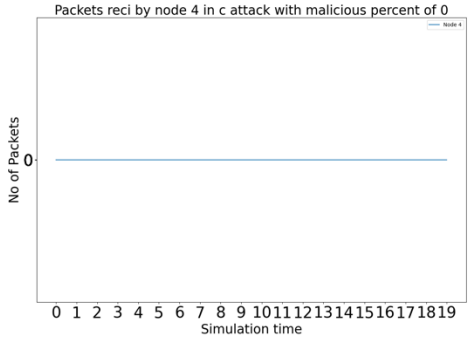

Fig 4.1.0.C.R.4: Packets Received by Node 4 with 0 attackers in Constant attack.

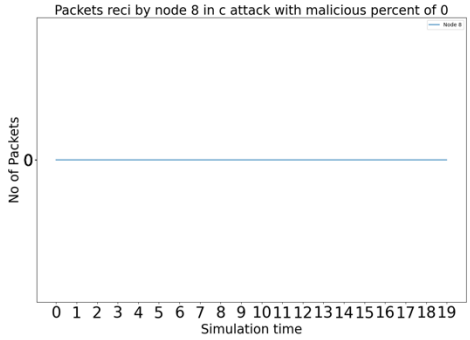

Fig 4.1.0.C.R.8: Packets Received by Node 8 with 0 attackers in Constant attack.

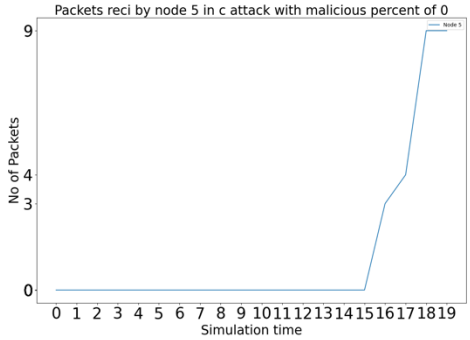

Fig 4.1.0.C.R.5: Packets Received by Node 5 with 0 attackers in Constant attack.

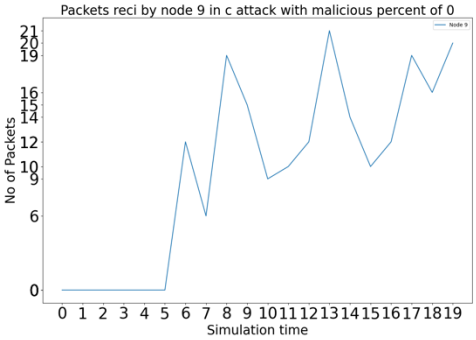

Fig 4.1.0.C.R.9: Packets Received by Node 9 with 0 attackers in Constant attack.

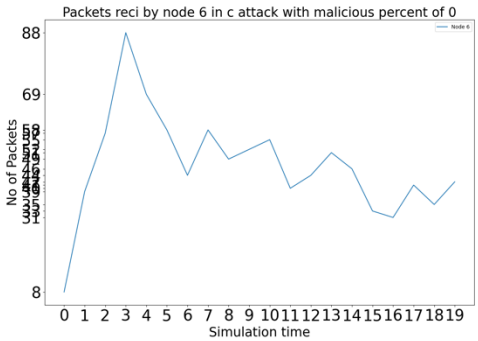

Fig 4.1.0.C.R.6: Packets Received by Node 6 with 0 attackers in Constant attack.

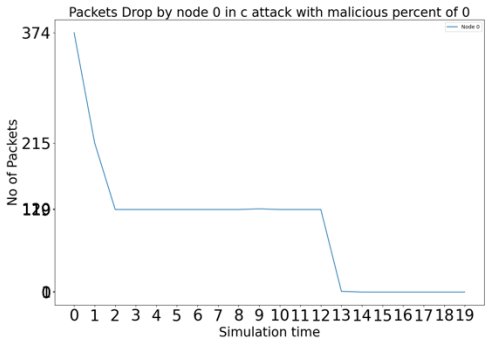

Fig 4.1.0.C.D.0: Packets Dropped by Node 0 with 0 attackers in Constant attack.

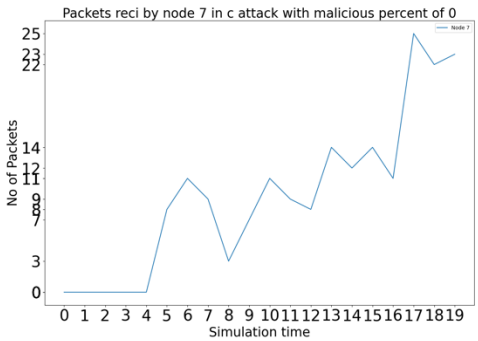

Fig 4.1.0.C.R.7: Packets Received by Node 7 with 0 attackers in Constant attack.

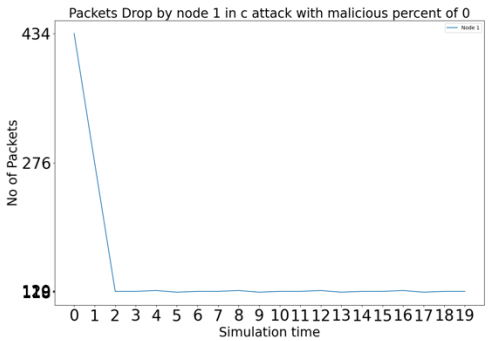

Fig 4.1.0.C.D.1: Packets Dropped by Node 1 with 0 attackers in Constant attack.

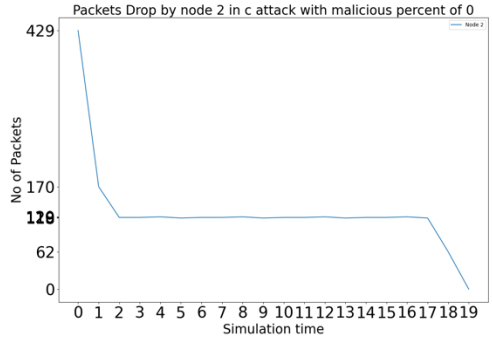

Fig 4.1.0.C.D.2: Packets Dropped by Node 2 with 0 attackers in Constant attack.

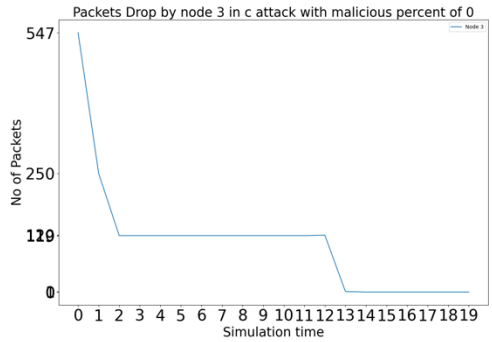

Fig 4.1.0.C.D.3: Packets Dropped by Node 3 with 0 attackers in Constant attack.

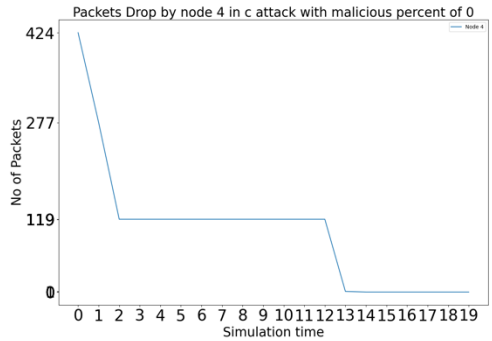

Fig 4.1.0.C.D.4: Packets Dropped by Node 4 with 0 attackers in Constant attack.

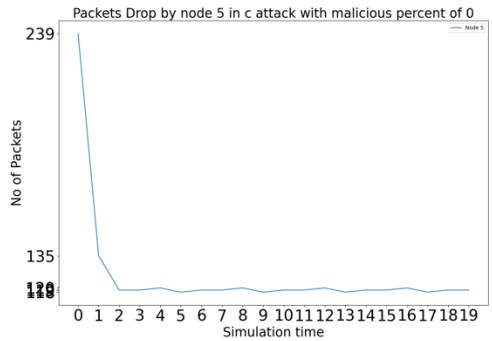

Fig 4.1.0.C.D.5: Packets Dropped by Node 5 with 0 attackers in Constant attack.

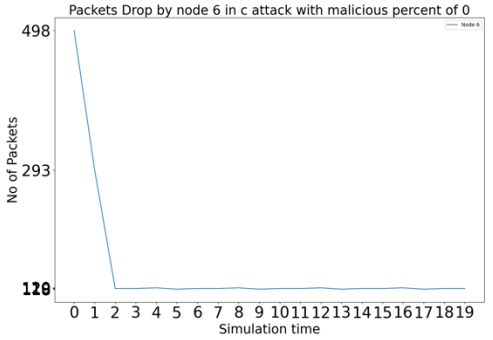

Fig 4.1.0.C.D.6: Packets Dropped by Node 6 with 0 attackers in Constant attack.

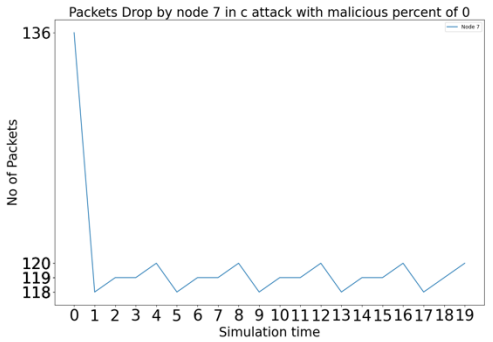

Fig 4.1.0.C.D.7: Packets Dropped by Node 7 with 0 attackers in Constant attack.

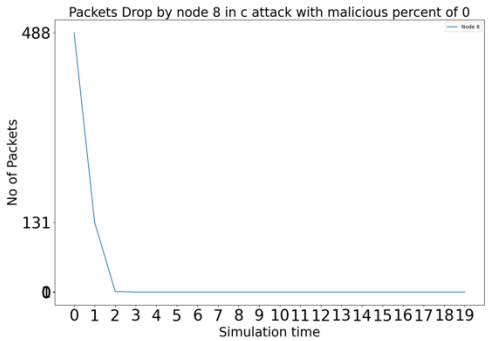

Fig 4.1.0.C.D.8: Packets Dropped by Node 8 with 0 attackers in Constant attack.

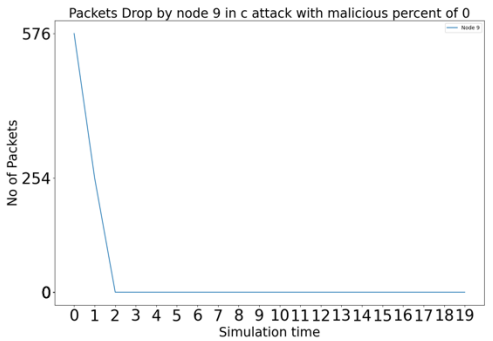

Fig 4.1.0.C.D.9: Packets Dropped by Node 9 with 0 attackers in Constant attack.

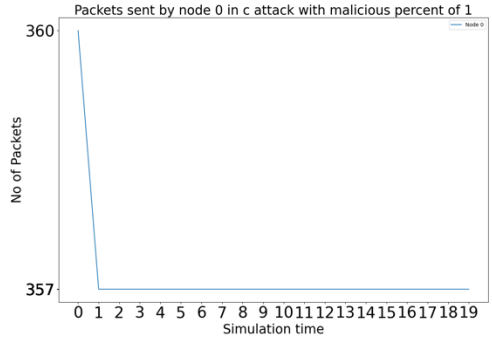

Fig 4.1.1.C.S.0: Packets Sent by Node 0 with 1 attacker in Constant attack.

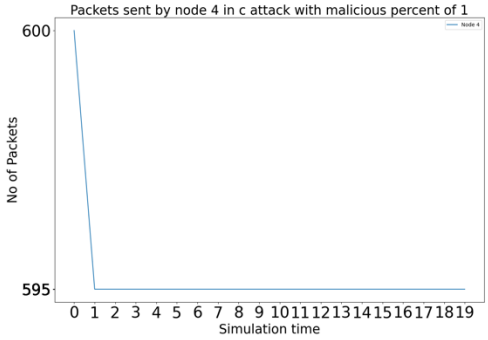

Fig 4.1.1.C.S.4: Packets Sent by Node 4 with 1 attacker in Constant attack.

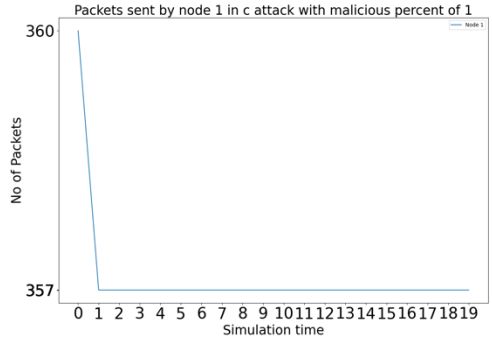

Fig 4.1.1.C.S.1: Packets Sent by Node 1 with 1 attacker in Constant attack.

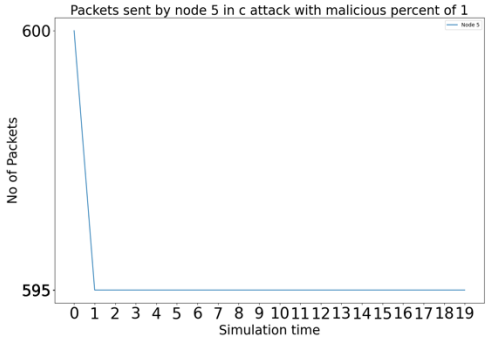

Fig 4.1.1.C.S.5: Packets Sent by Node 5 with 1 attacker in Constant attack.

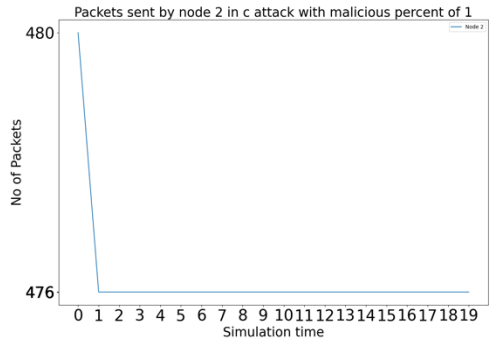

Fig 4.1.1.C.S.2: Packets Sent by Node 2 with 1 attacker in Constant attack.

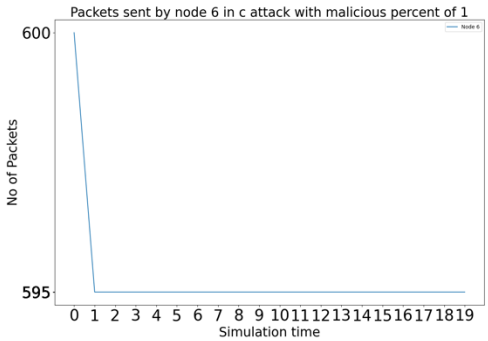

Fig 4.1.1.C.S.6: Packets Sent by Node 6 with 1 attacker in Constant attack.

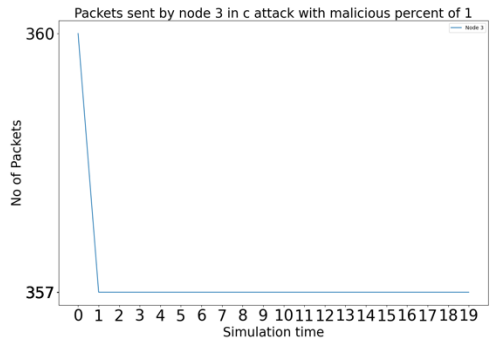

Fig 4.1.1.C.S.3: Packets Sent by Node 3 with 1 attacker in Constant attack.

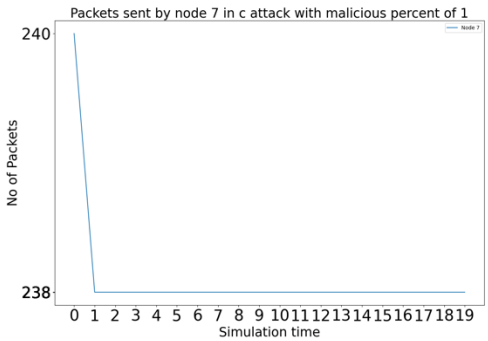

Fig 4.1.1.C.S.7: Packets Sent by Node 7 with 1 attacker in Constant attack.

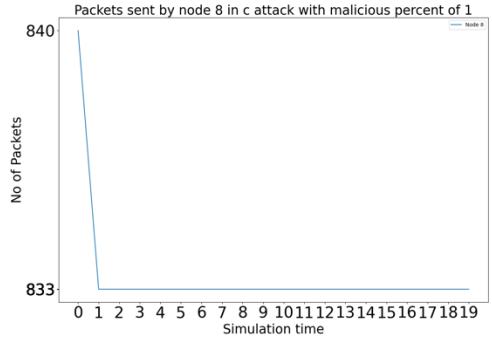

Fig 4.1.1.C.S.8: Packets Sent by Node 8 with 1 attacker in Constant attack.

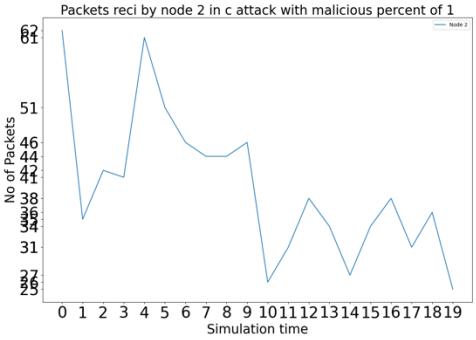

Fig 4.1.1.C.R.2: Packets Received by Node 2 with 1 attacker in Constant attack.

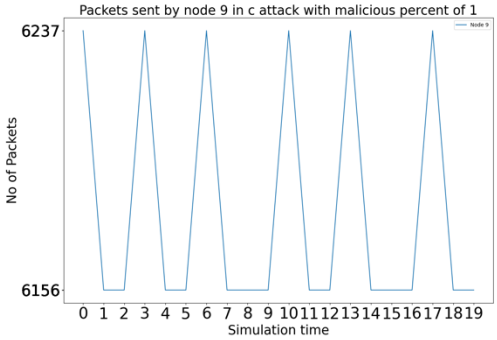

Fig 4.1.1.C.S.9: Packets Sent by Node 9 with 1 attacker in Constant attack.

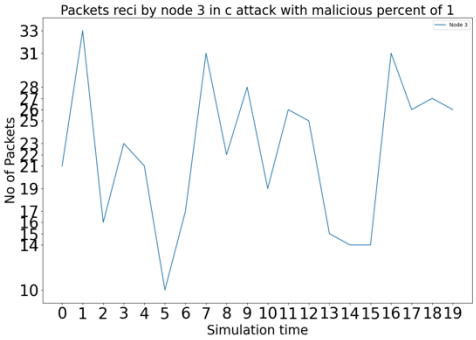

Fig 4.1.1.C.R.3: Packets Received by Node 3 with 1 attacker in Constant attack.

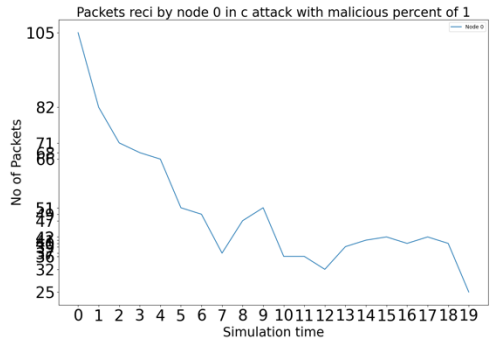

Fig 4.1.1.C.R.0: Packets Received by Node 0 with 1 attacker in Constant attack.

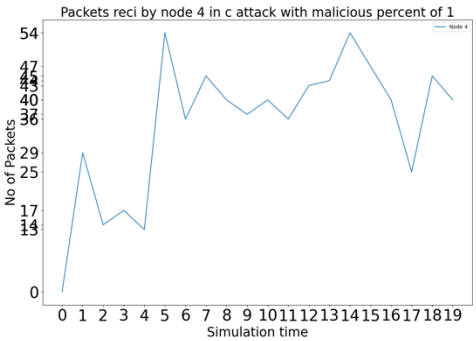

Fig 4.1.1.C.R.4: Packets Received by Node 4 with 1 attacker in Constant attack.

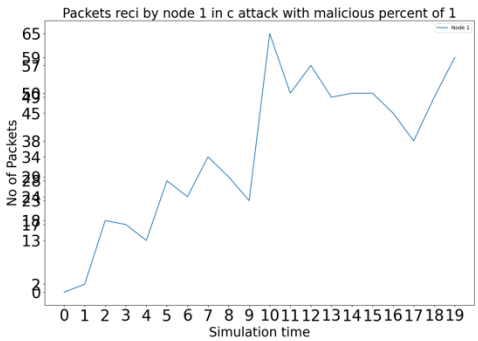

Fig 4.1.1.C.R.1: Packets Received by Node 1 with 1 attacker in Constant attack.

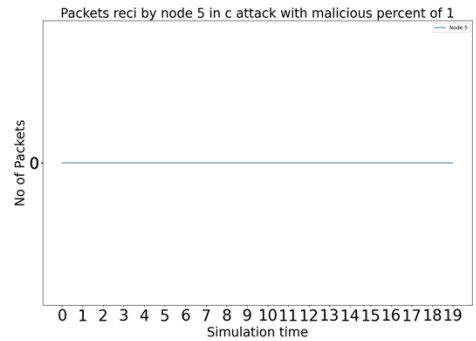

Fig 4.1.1.C.R.5: Packets Received by Node 5 with 1 attacker in Constant attack.

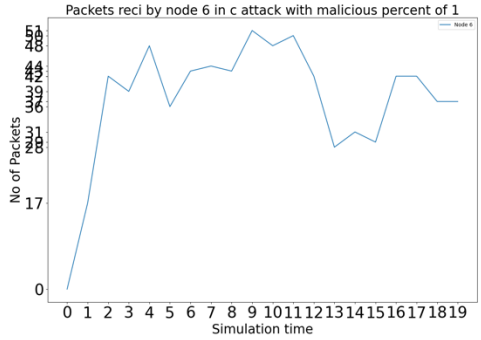

Fig 4.1.1.C.R.6: Packets Received by Node 6 with 1 attacker in Constant attack.

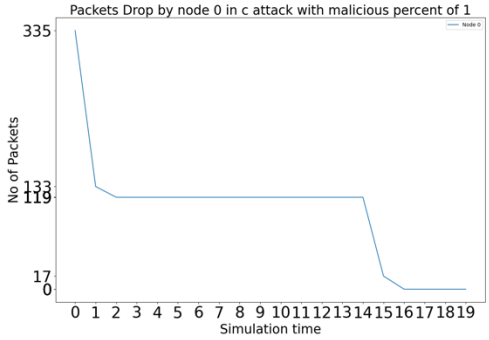

Fig 4.1.1.C.D.0: Packets Dropped by Node 0 with 1 attacker in Constant attack.

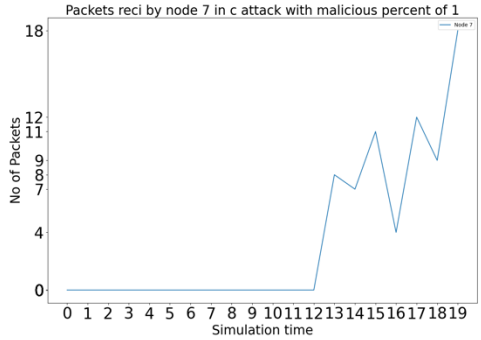

Fig 4.1.1.C.R.7: Packets Received by Node 7 with 1 attacker in Constant attack.

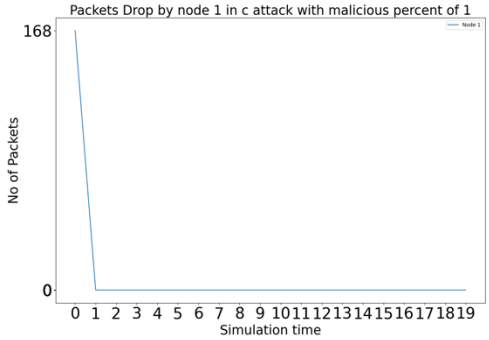

Fig 4.1.1.C.D.1: Packets Dropped by Node 1 with 1 attacker in Constant attack.

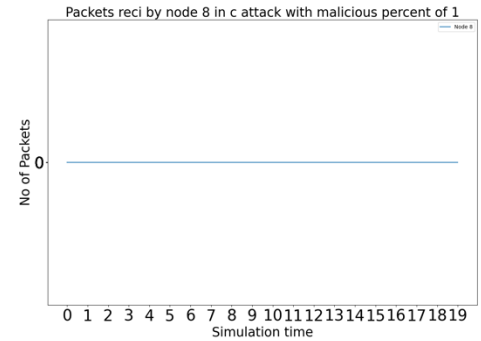

Fig 4.1.1.C.R.8: Packets Received by Node 8 with 1 attacker in Constant attack.

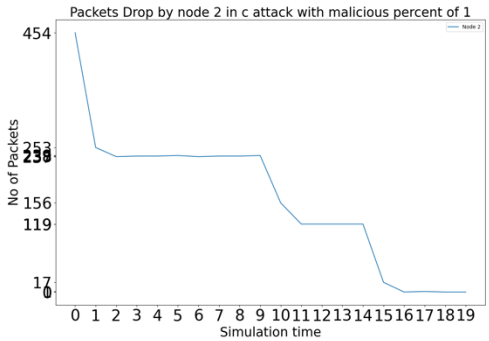

Fig 4.1.1.C.D.2: Packets Dropped by Node 2 with 1 attacker in Constant attack.

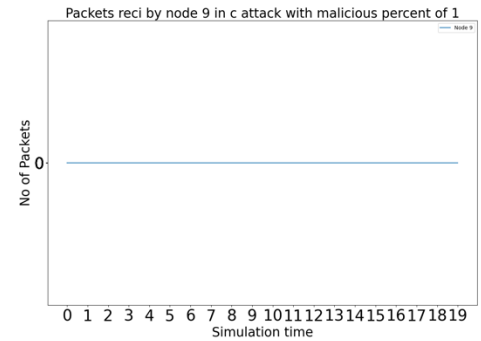

Fig 4.1.1.C.R.9: Packets Received by Node 9 with 1 attacker in Constant attack.

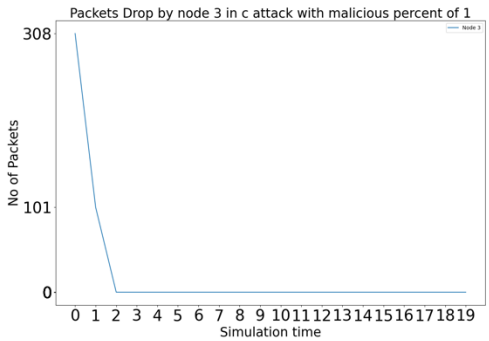

Fig 4.1.1.C.D.3: Packets Dropped by Node 3 with 1 attacker in Constant attack.

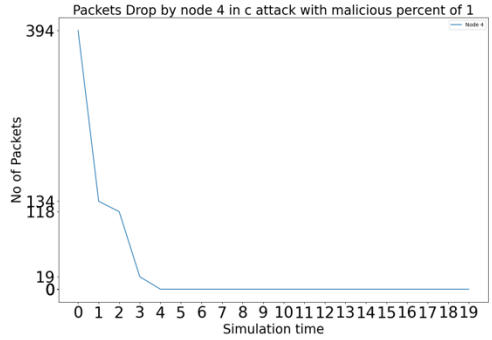

Fig 4.1.1.C.D.4: Packets Dropped by Node 4 with 1 attacker in Constant attack.

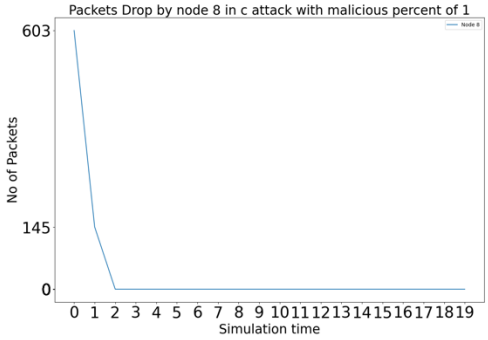

Fig 4.1.1.C.D.8: Packets Dropped by Node 8 with 1 attacker in Constant attack.

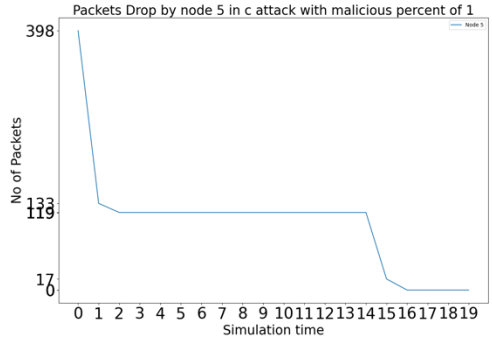

Fig 4.1.1.C.D.5: Packets Dropped by Node 5 with 1 attacker in Constant attack.

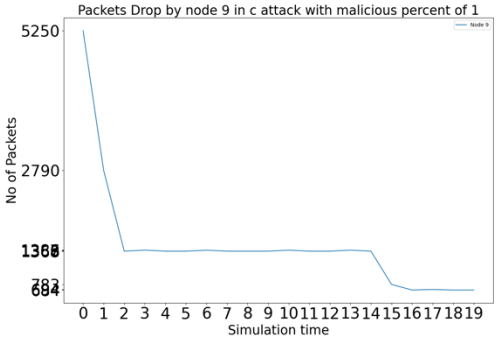

Fig 4.1.1.C.D.9: Packets Dropped by Node 9 with 1 attacker in Constant attack.

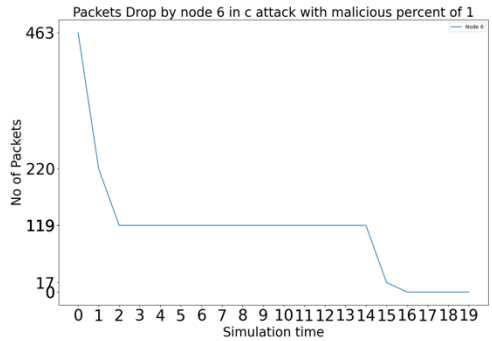

Fig 4.1.1.C.D.6: Packets Dropped by Node 6 with 1 attacker in Constant attack.

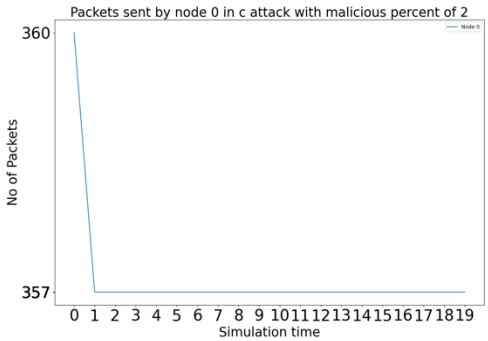

Fig 4.1.2.C.S.0: Packets Sent by Node 0 with 2 attackers in Constant attack.

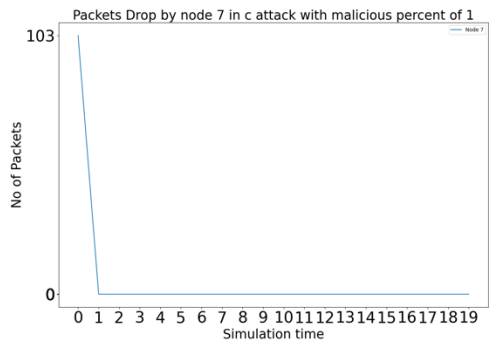

Fig 4.1.1.C.D.7: Packets Dropped by Node 7 with 1 attacker in Constant attack.

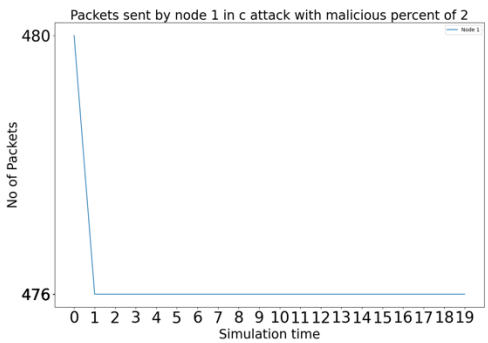

Fig 4.1.2.C.S.1: Packets Sent by Node 1 with 2 attackers in Constant attack.

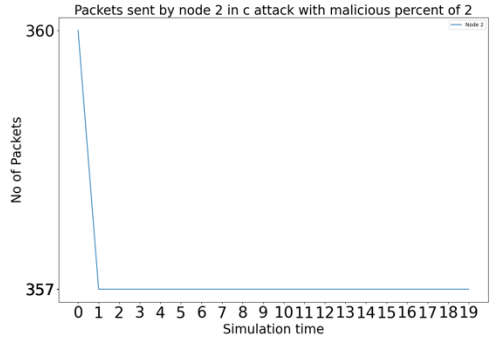

Fig 4.1.2.C.S.2: Packets Sent by Node 2 with 2 attackers in Constant attack.

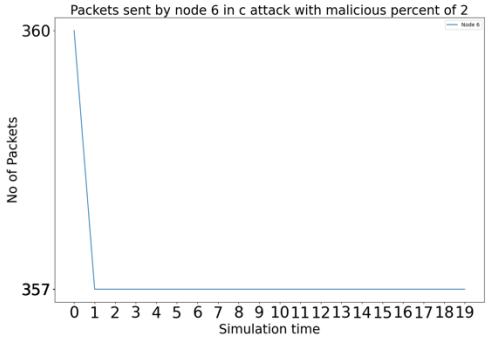

Fig 4.1.2.C.S.6: Packets Sent by Node 6 with 2 attackers in Constant attack.

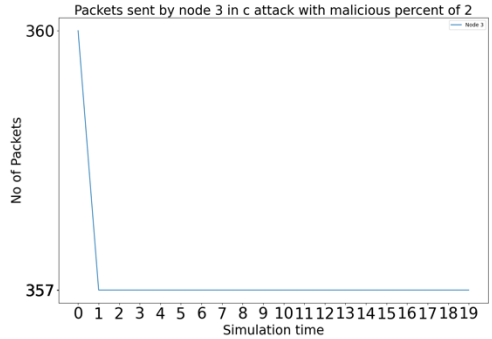

Fig 4.1.2.C.S.3: Packets Sent by Node 3 with 2 attackers in Constant attack.

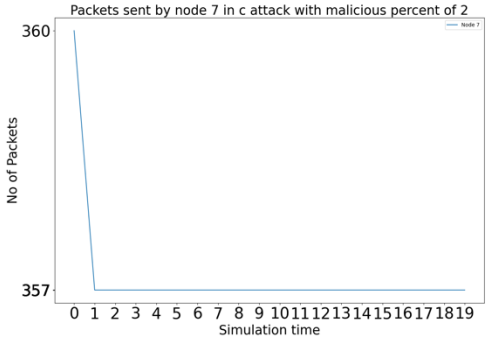

Fig 4.1.2.C.S.7: Packets Sent by Node 7 with 2 attackers in Constant attack.

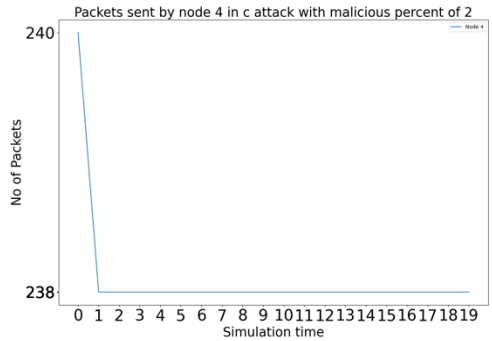

Fig 4.1.2.C.S.4: Packets Sent by Node 4 with 2 attackers in Constant attack.

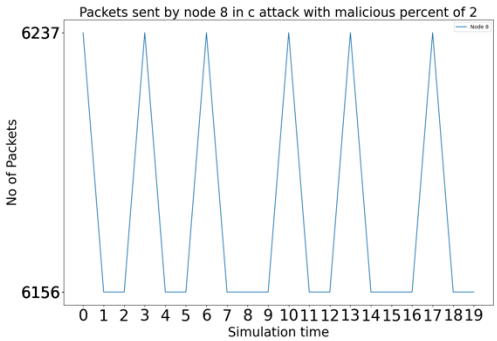

Fig 4.1.2.C.S.8: Packets Sent by Node 8 with 2 attackers in Constant attack.

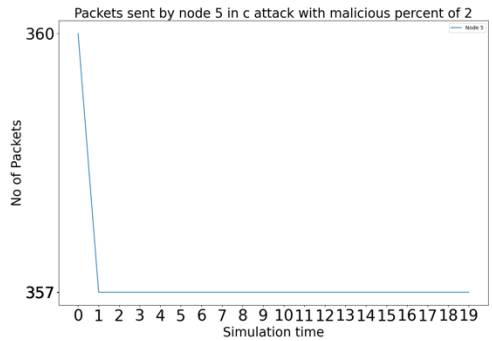

Fig 4.1.2.C.S.5: Packets Sent by Node 5 with 2 attackers in Constant attack.

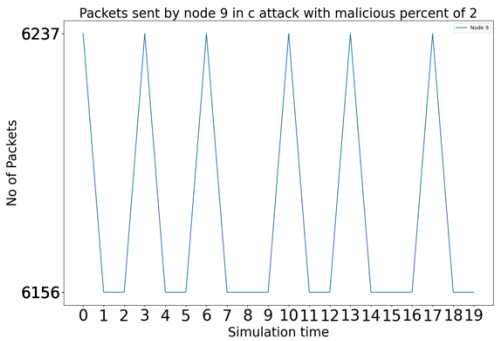

Fig 4.1.2.C.S.9: Packets Sent by Node 9 with 2 attackers in Constant attack.

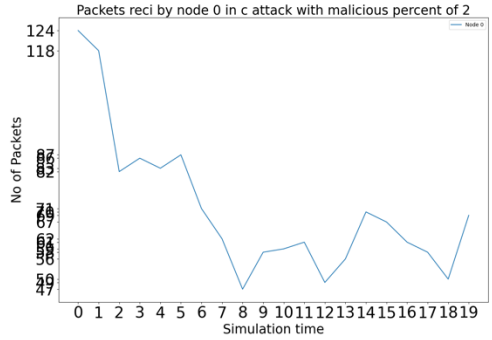

Fig 4.1.2.C.R.0: Packets Received by Node 0 with 2 attackers in Constant attack.

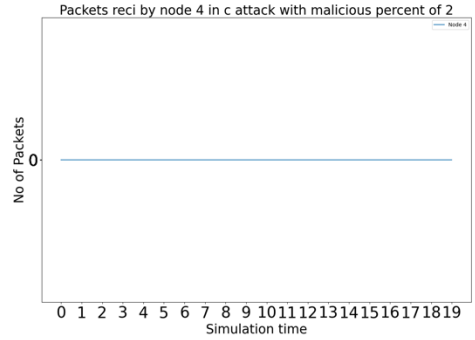

Fig 4.1.2.C.R.4: Packets Received by Node 4 with 2 attackers in Constant attack.

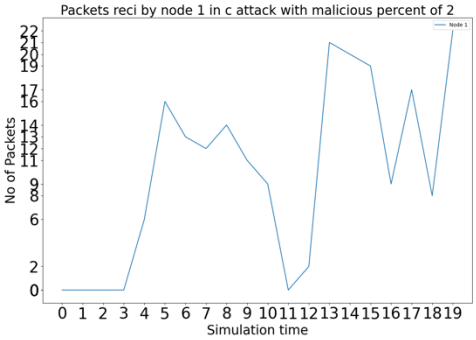

Fig 4.1.2.C.R.1: Packets Received by Node 1 with 2 attackers in Constant attack.

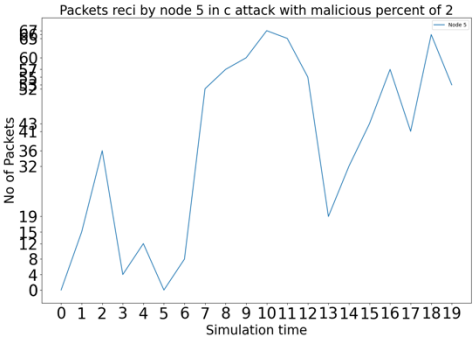

Fig 4.1.2.C.R.5: Packets Received by Node 5 with 2 attackers in Constant attack.

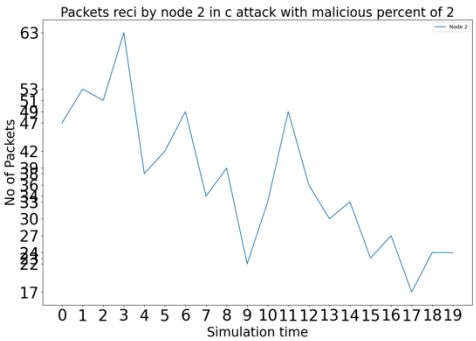

Fig 4.1.2.C.R.2: Packets Received by Node 2 with 2 attackers in Constant attack.

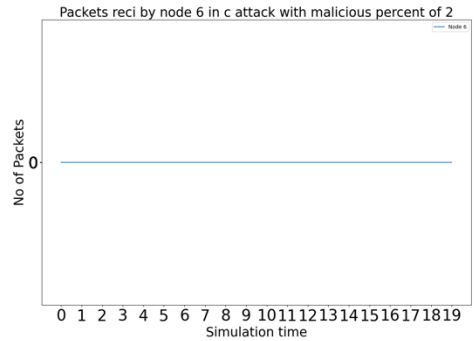

Fig 4.1.2.C.R.6: Packets Received by Node 6 with 2 attackers in Constant attack.

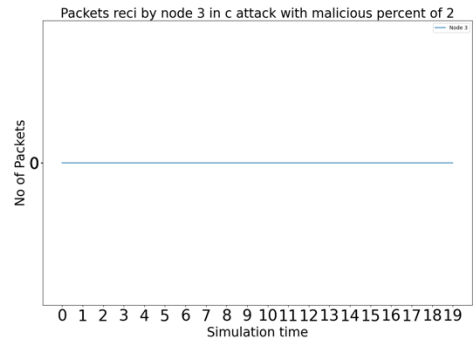

Fig 4.1.2.C.R.3: Packets Received by Node 3 with 2 attackers in Constant attack.

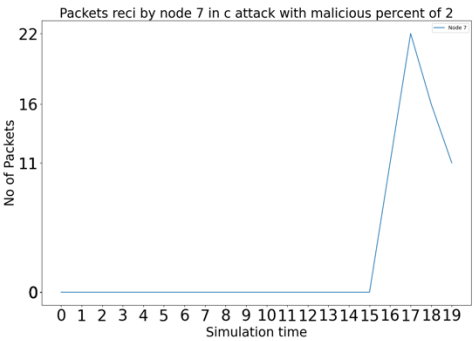

Fig 4.1.2.C.R.7: Packets Received by Node 7 with 2 attackers in Constant attack.

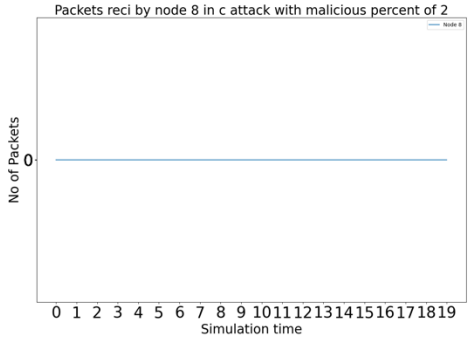

Fig 4.1.2.C.R.8: Packets Received by Node 8 with 2 attackers in Constant attack.

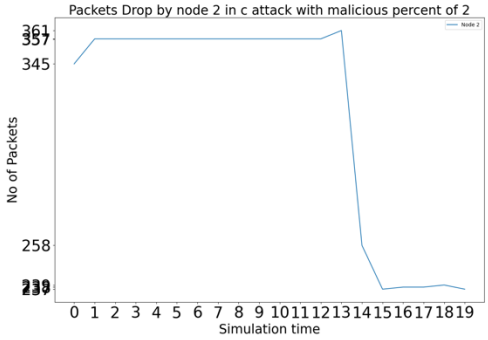

Fig 4.1.2.C.D.2: Packets Dropped by Node 2 with 2 attackers in Constant attack.

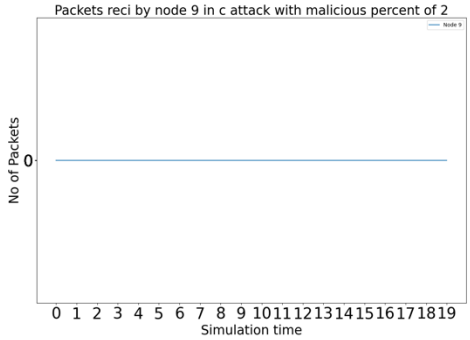

Fig 4.1.2.C.R.9: Packets Received by Node 9 with 2 attackers in Constant attack.

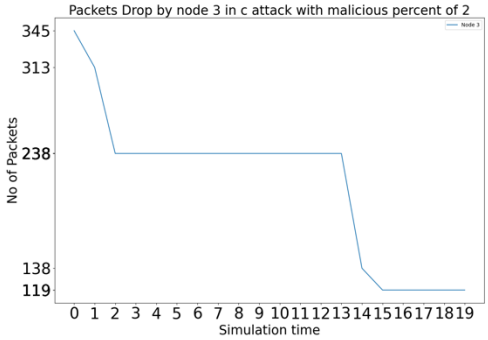

Fig 4.1.2.C.D.3: Packets Dropped by Node 3 with 2 attackers in Constant attack.

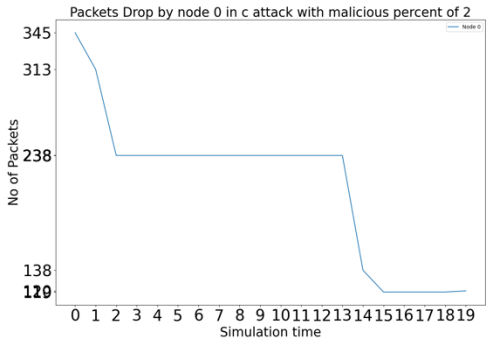

Fig 4.1.2.C.D.0: Packets Dropped by Node 0 with 2 attackers in Constant attack.

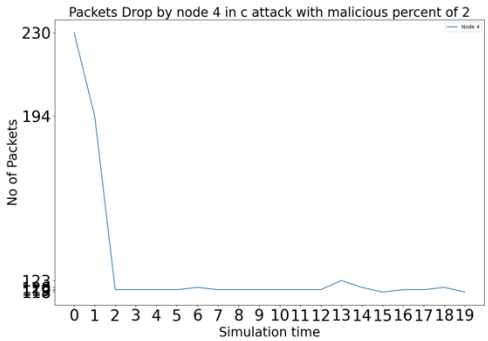

Fig 4.1.2.C.D.4: Packets Dropped by Node 4 with 2 attackers in Constant attack.

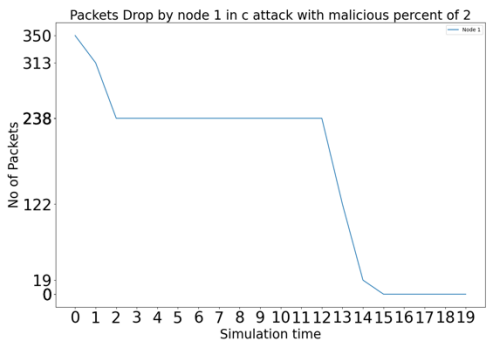

Fig 4.1.2.C.D.1: Packets Dropped by Node 1 with 2 attackers in Constant attack.

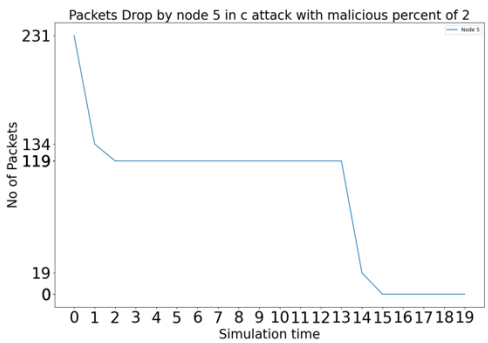

Fig 4.1.2.C.D.5: Packets Dropped by Node 5 with 2 attackers in Constant attack.

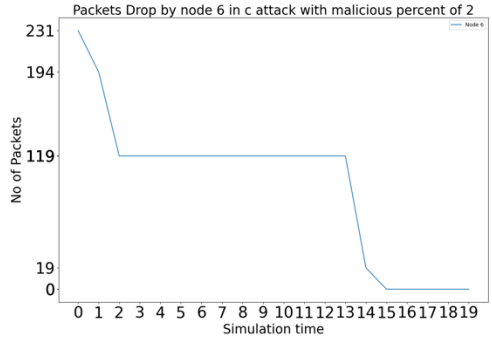

Fig 4.1.2.C.D.6: Packets Dropped by Node 6 with 2 attackers in Constant attack.

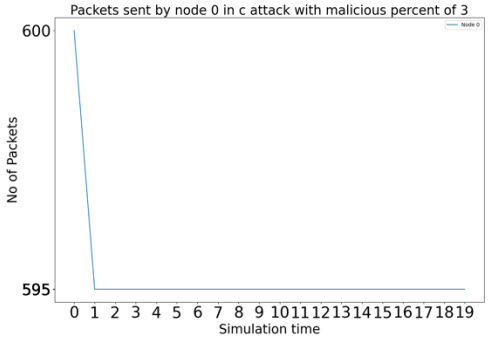

Fig 4.1.3.C.S.0: Packets Sent by Node 0 with 3 attackers in Constant attack.

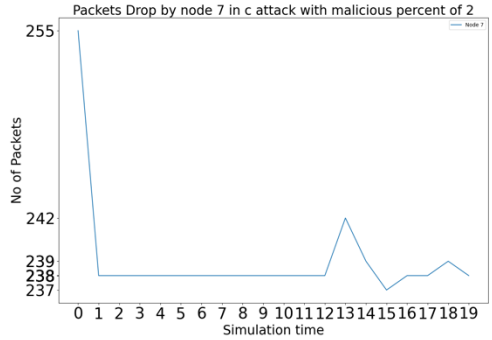

Fig 4.1.2.C.D.7: Packets Dropped by Node 7 with 2 attackers in Constant attack.

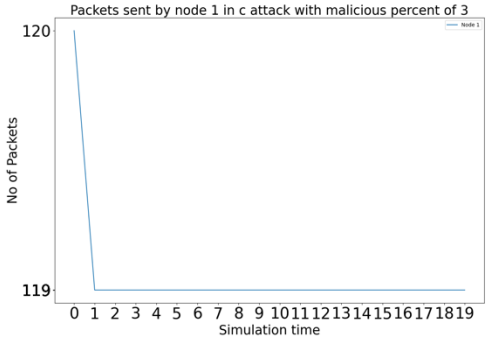

Fig 4.1.3.C.S.1: Packets Sent by Node 1 with 3 attackers in Constant attack.

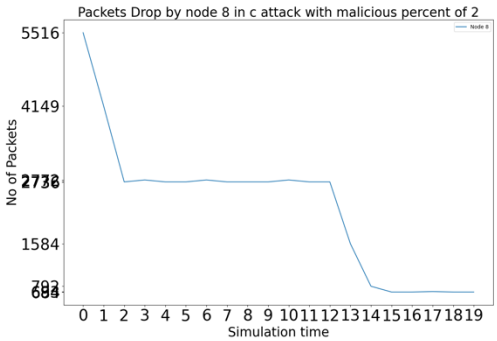

Fig 4.1.2.C.D.8: Packets Dropped by Node 8 with 2 attackers in Constant attack.

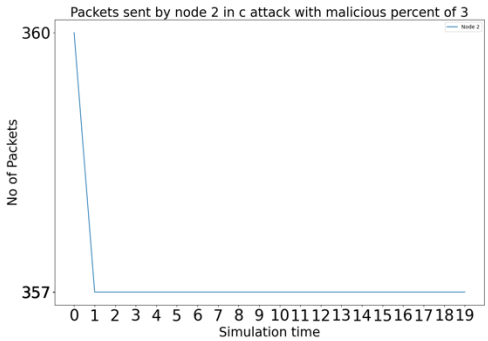

Fig 4.1.3.C.S.2: Packets Sent by Node 2 with 3 attackers in Constant attack.

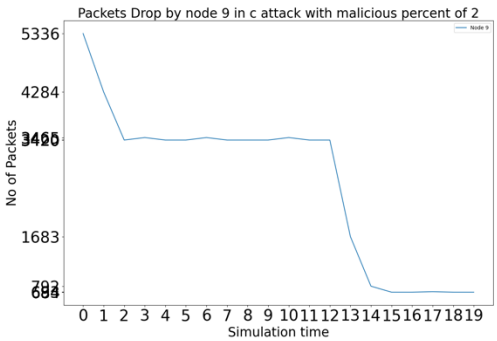

Fig 4.1.2.C.D.9: Packets Dropped by Node 9 with 2 attackers in Constant attack.

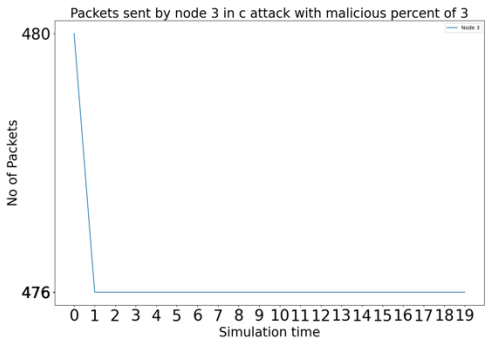

Fig 4.1.3.C.S.3: Packets Sent by Node 3 with 3 attackers in Constant attack.

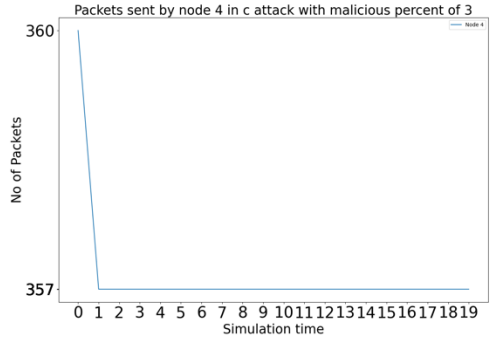

Fig 4.1.3.C.S.4: Packets Sent by Node 4 with 3 attackers in Constant attack.

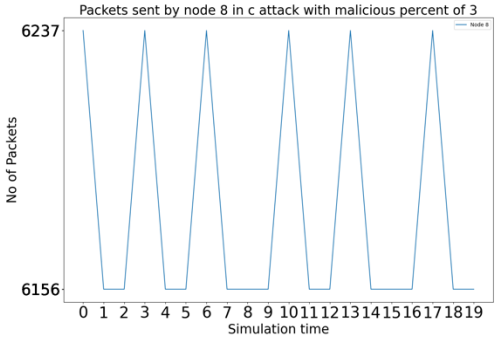

Fig 4.1.3.C.S.8: Packets Sent by Node 8 with 3 attackers in Constant attack.

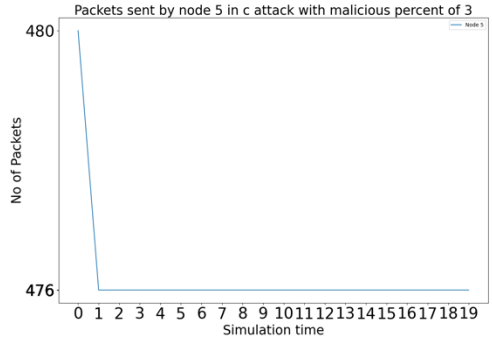

Fig 4.1.3.C.S.5: Packets Sent by Node 5 with 3 attackers in Constant attack.

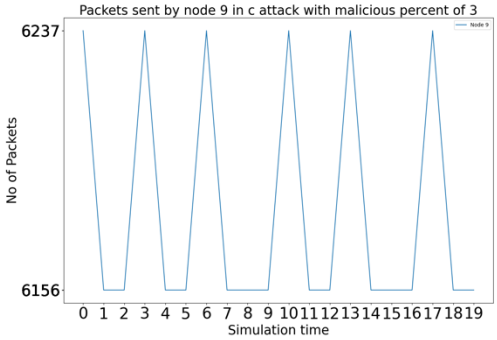

Fig 4.1.3.C.S.9: Packets Sent by Node 9 with 3 attackers in Constant attack.

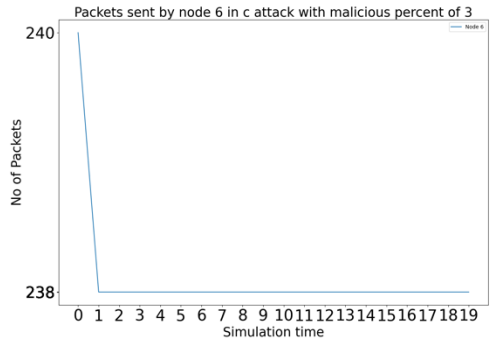

Fig 4.1.3.C.S.6: Packets Sent by Node 6 with 3 attackers in Constant attack.

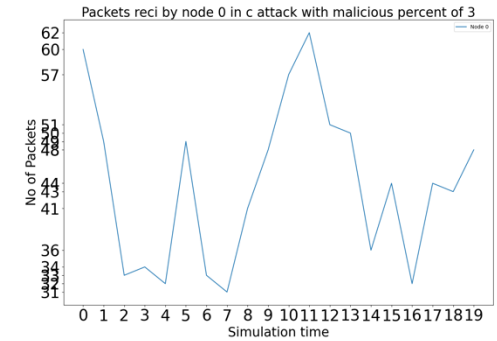

Fig 4.1.3.C.R.0: Packets Received by Node 0 with 3 attackers in Constant attack.

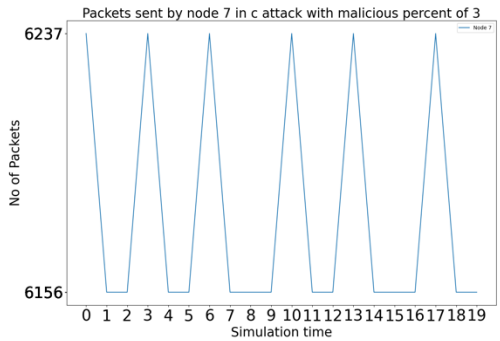

Fig 4.1.3.C.S.7: Packets Sent by Node 7 with 3 attackers in Constant attack.

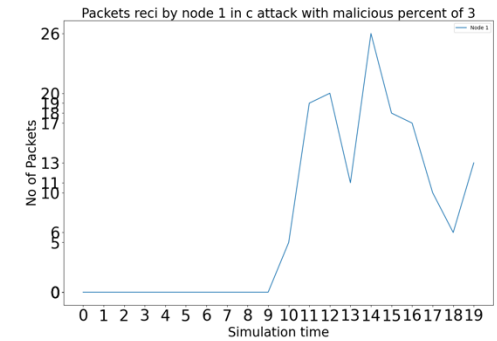

Fig 4.1.3.C.R.1: Packets Received by Node 1 with 3 attackers in Constant attack.

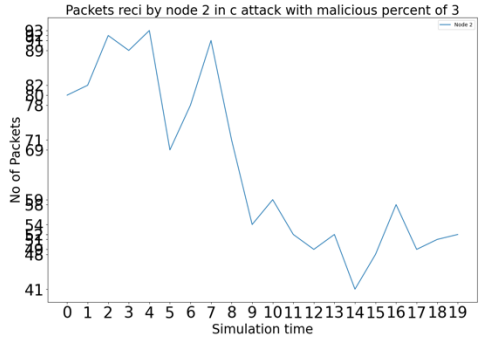

Fig 4.1.3.C.R.2: Packets Received by Node 2 with 3 attackers in Constant attack.

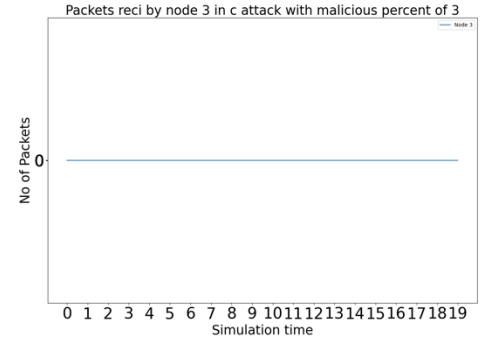

Fig 4.1.3.C.R.3: Packets Received by Node 3 with 3 attackers in Constant attack.

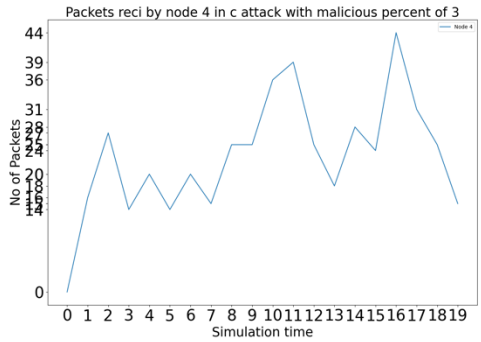

Fig 4.1.3.C.R.4: Packets Received by Node 4 with 3 attackers in Constant attack.

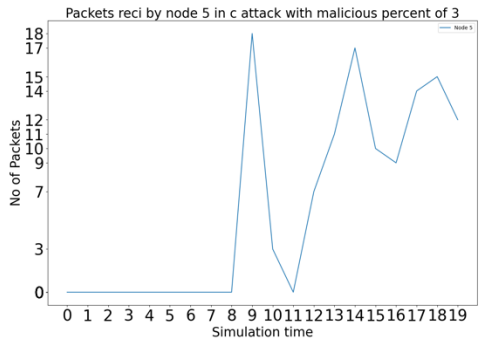

Fig 4.1.3.C.R.5: Packets Received by Node 5 with 3 attackers in Constant attack.

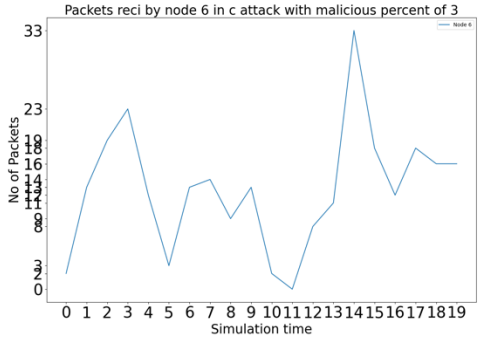

Fig 4.1.3.C.R.6: Packets Received by Node 6 with 3 attackers in Constant attack.

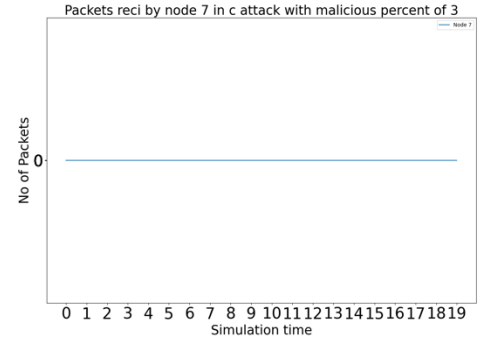

Fig 4.1.3.C.R.7: Packets Received by Node 7 with 3 attackers in Constant attack.

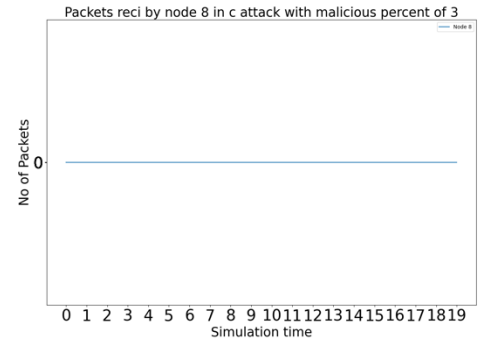

Fig 4.1.3.C.R.8: Packets Received by Node 8 with 3 attackers in Constant attack.

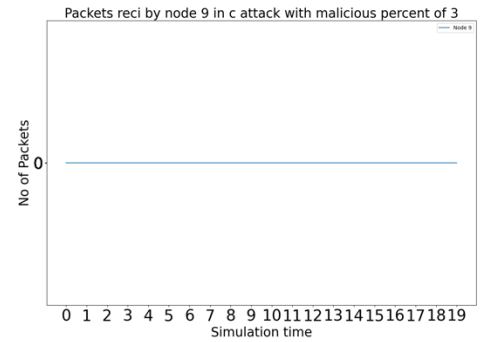

Fig 4.1.3.C.R.9: Packets Received by Node 9 with 3 attackers in Constant attack.

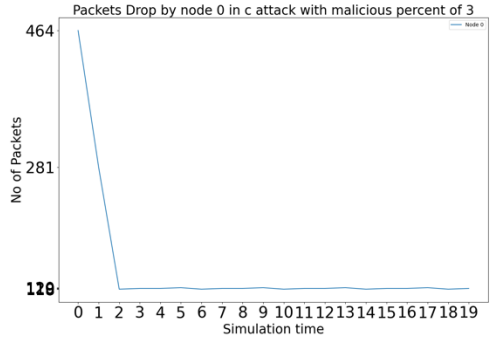

Fig 4.1.3.C.D.0: Packets Dropped by Node 0 with 3 attackers in Constant attack.

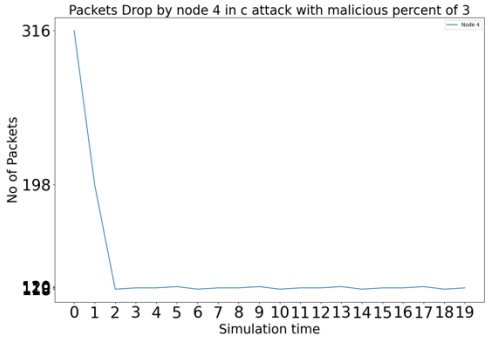

Fig 4.1.3.C.D.4: Packets Dropped by Node 4 with 3 attackers in Constant attack.

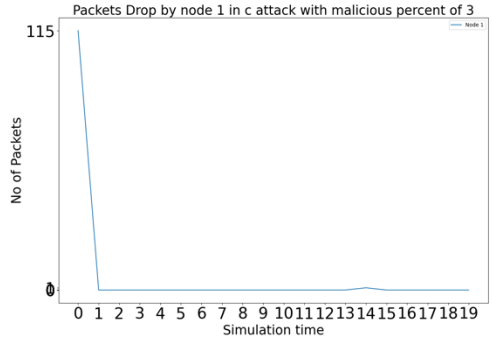

Fig 4.1.3.C.D.1: Packets Dropped by Node 1 with 3 attackers in Constant attack.

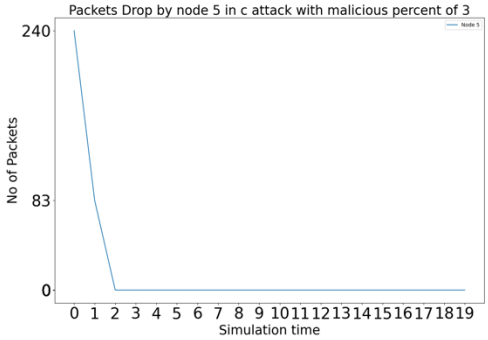

Fig 4.1.3.C.D.5: Packets Dropped by Node 5 with 3 attackers in Constant attack.

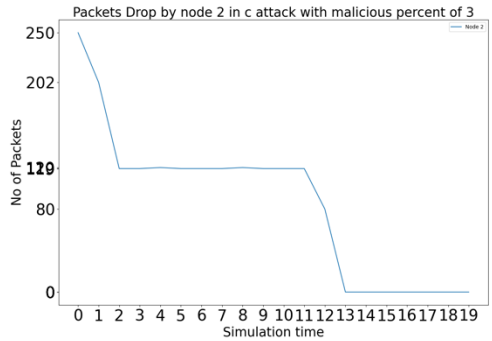

Fig 4.1.3.C.D.2: Packets Dropped by Node 2 with 3 attackers in Constant attack.

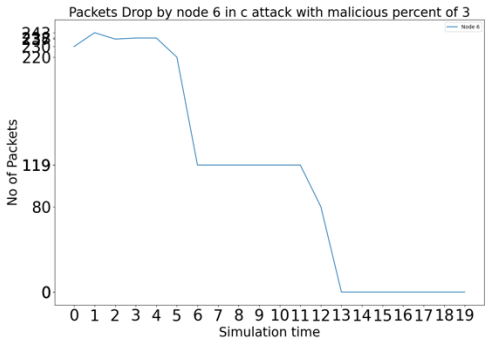

Fig 4.1.3.C.D.6: Packets Dropped by Node 6 with 3 attackers in Constant attack.

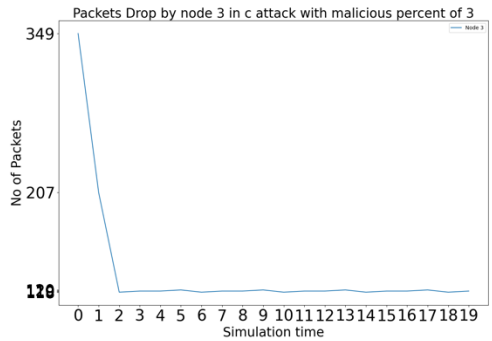

Fig 4.1.3.C.D.3: Packets Dropped by Node 3 with 3 attackers in Constant attack.

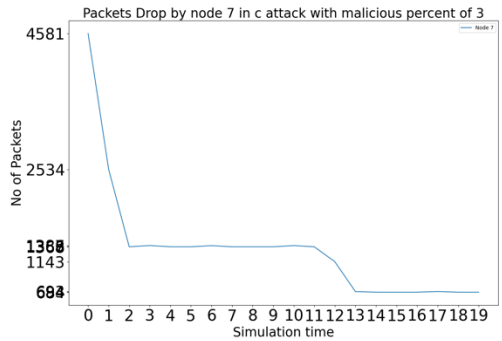

Fig 4.1.3.C.D.7: Packets Dropped by Node 7 with 3 attackers in Constant attack.

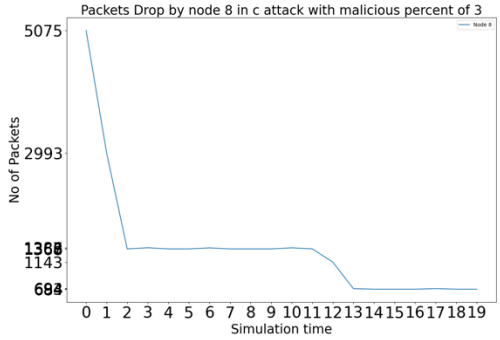

Fig 4.1.3.C.D.8: Packets Dropped by Node 8 with 3 attackers in Constant attack.

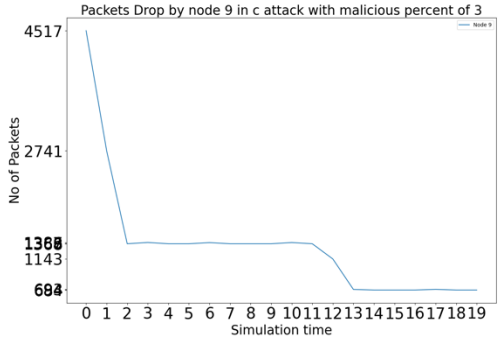

Fig 4.1.3.C.D.9: Packets Dropped by Node 9 with 3 attackers in Constant attack.

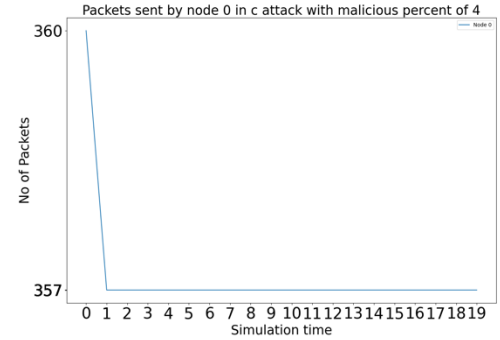

Fig 4.1.4.C.S.0: Packets Sent by Node 0 with 4 attackers in Constant attack.

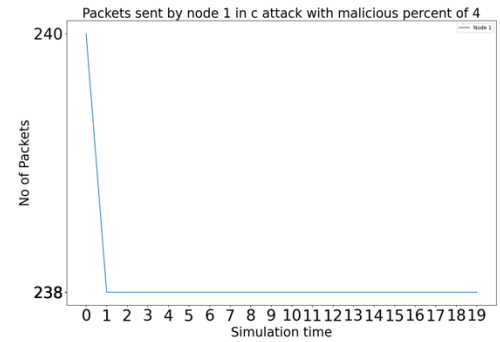

Fig 4.1.4.C.S.1: Packets Sent by Node 1 with 4 attackers in Constant attack.

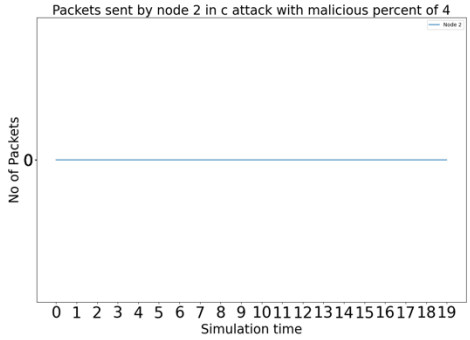

Fig 4.1.4.C.S.2: Packets Sent by Node 2 with 4 attackers in Constant attack.

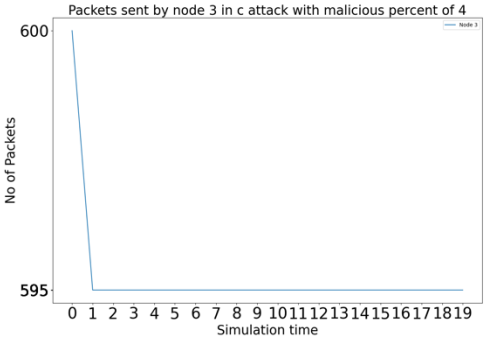

Fig 4.1.4.C.S.3: Packets Sent by Node 3 with 4 attackers in Constant attack.

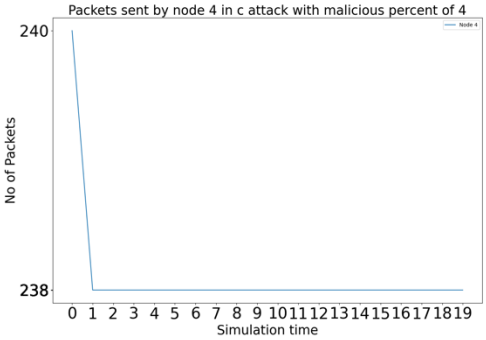

Fig 4.1.4.C.S.4: Packets Sent by Node 4 with 4 attackers in Constant attack.

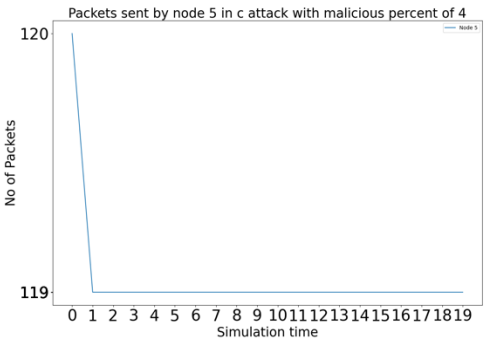

Fig 4.1.4.C.S.5: Packets Sent by Node 5 with 4 attackers in Constant attack.

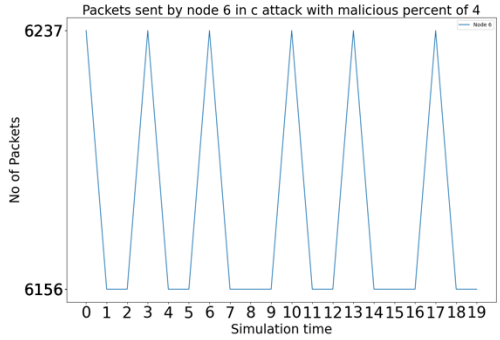

Fig 4.1.4.C.S.6: Packets Sent by Node 6 with 4 attackers in Constant attack.

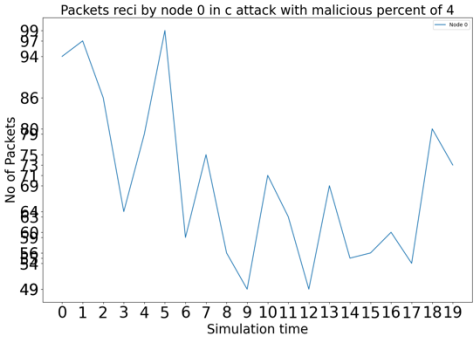

Fig 4.1.4.C.R.0: Packets Received by Node 0 with 4 attackers in Constant attack.

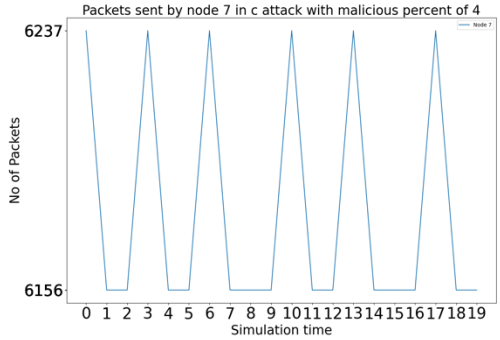

Fig 4.1.4.C.S.7: Packets Sent by Node 7 with 4 attackers in Constant attack.

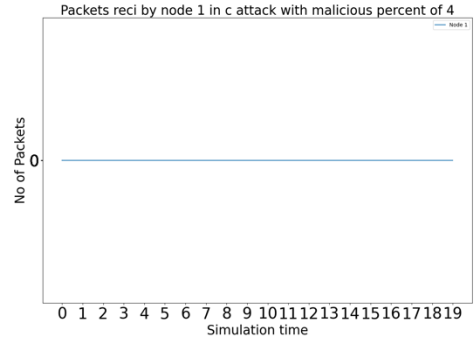

Fig 4.1.4.C.R.1: Packets Received by Node 1 with 4 attackers in Constant attack.

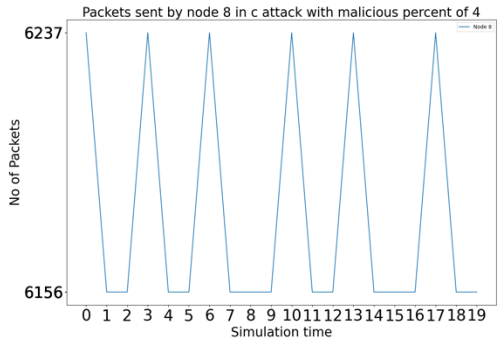

Fig 4.1.4.C.S.8: Packets Sent by Node 8 with 4 attackers in Constant attack.

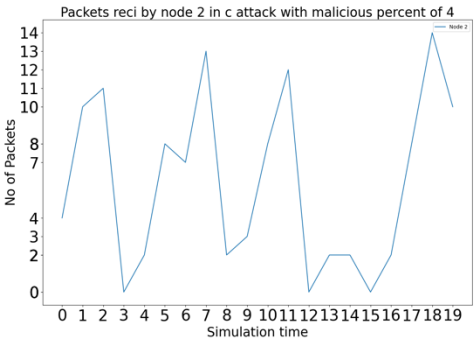

Fig 4.1.4.C.R.2: Packets Received by Node 2 with 4 attackers in Constant attack.

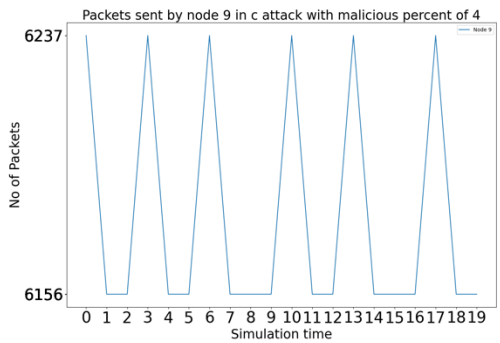

Fig 4.1.4.C.S.9: Packets Sent by Node 9 with 4 attackers in Constant attack.

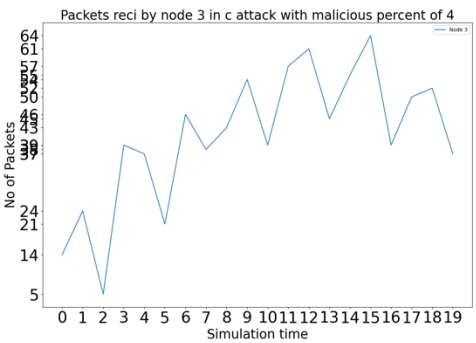

Fig 4.1.4.C.R.3: Packets Received by Node 3 with 4 attackers in Constant attack.

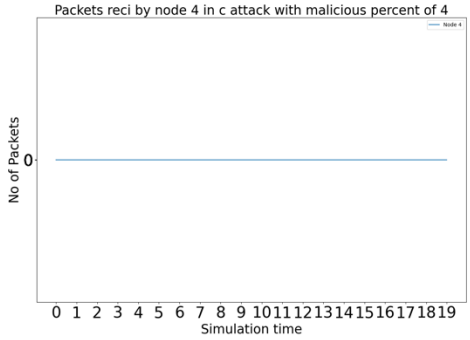

Fig 4.1.4.C.R.4: Packets Received by Node 4 with 4 attackers in Constant attack.

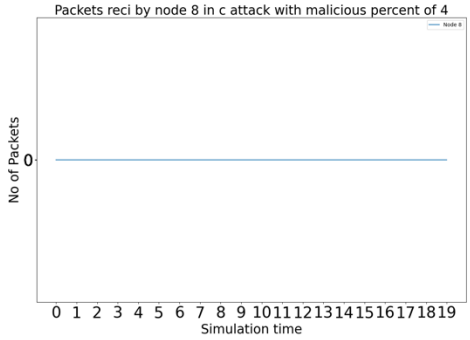

Fig 4.1.4.C.R.8: Packets Received by Node 8 with 4 attackers in Constant attack.

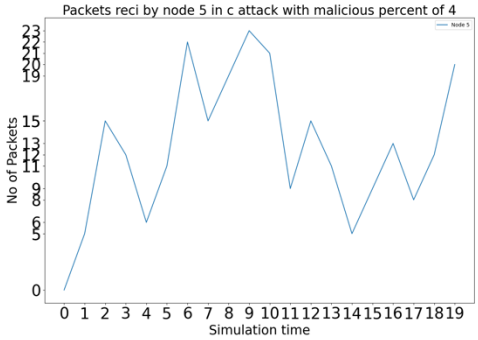

Fig 4.1.4.C.R.5: Packets Received by Node 5 with 4 attackers in Constant attack.

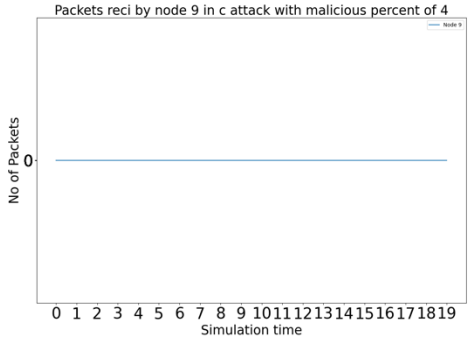

Fig 4.1.4.C.R.9: Packets Received by Node 9 with 4 attackers in Constant attack.

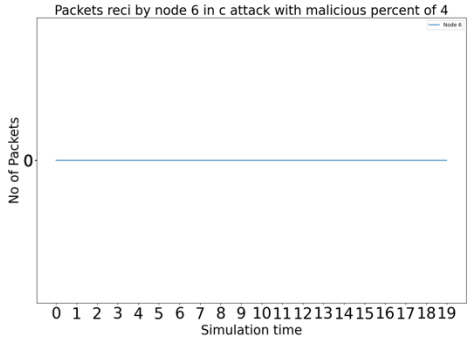

Fig 4.1.4.C.R.6: Packets Received by Node 6 with 4 attackers in Constant attack.

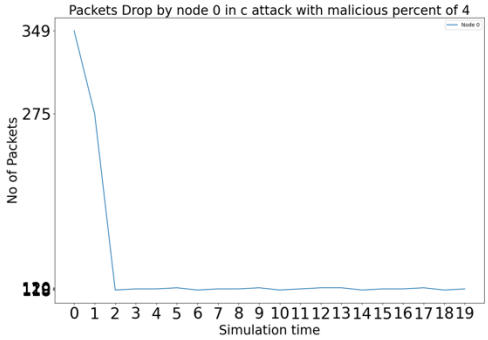

Fig 4.1.4.C.D.0: Packets Dropped by Node 0 with 4 attackers in Constant attack.

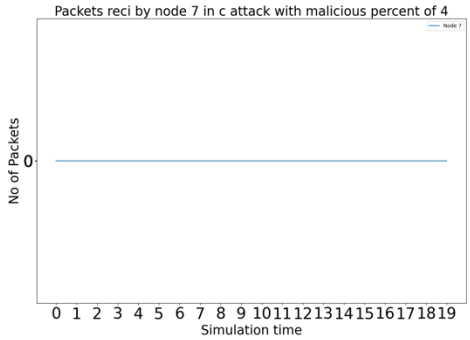

Fig 4.1.4.C.R.7: Packets Received by Node 7 with 4 attackers in Constant attack.

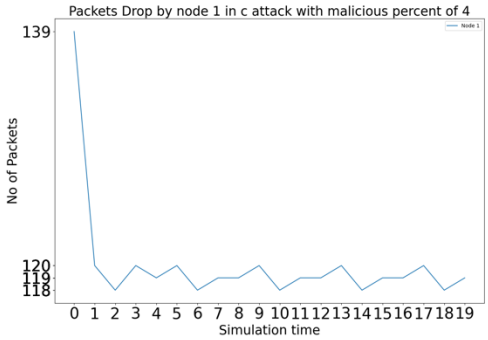

Fig 4.1.4.C.D.1: Packets Dropped by Node 1 with 4 attackers in Constant attack.

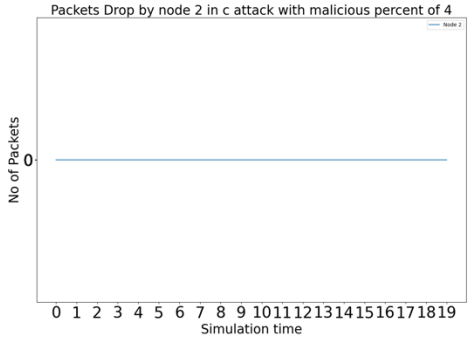

Fig 4.1.4.C.D.2: Packets Dropped by Node 2 with 4 attackers in Constant attack.

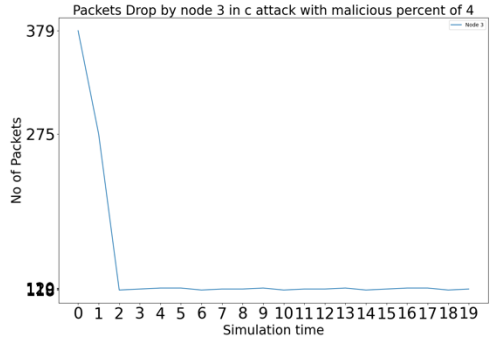

Fig 4.1.4.C.D.3: Packets Dropped by Node 3 with 4 attackers in Constant attack.

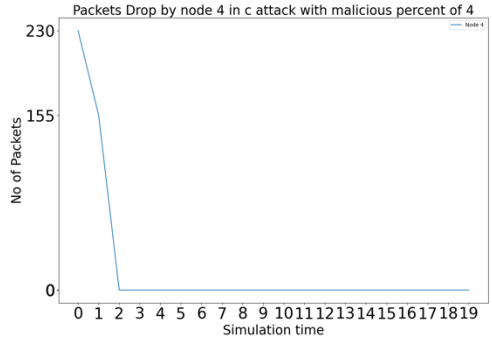

Fig 4.1.4.C.D.4: Packets Dropped by Node 4 with 4 attackers in Constant attack.

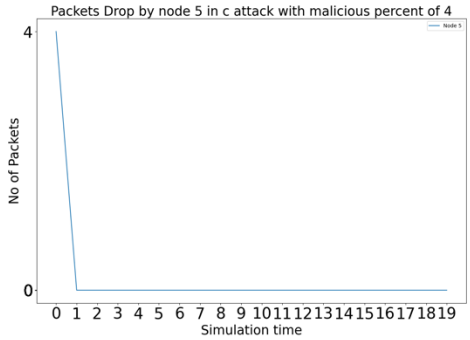

Fig 4.1.4.C.D.5: Packets Dropped by Node 5 with 4 attackers in Constant attack.

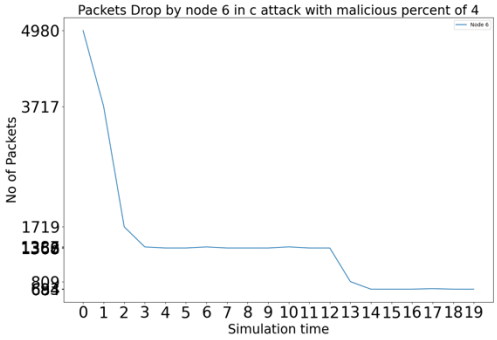

Fig 4.1.4.C.D.6: Packets Dropped by Node 6 with 4 attackers in Constant attack.

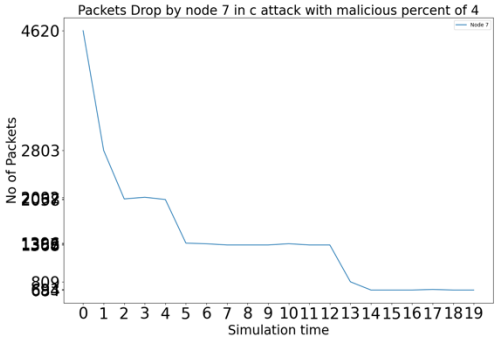

Fig 4.1.4.C.D.7: Packets Dropped by Node 7 with 4 attackers in Constant attack.

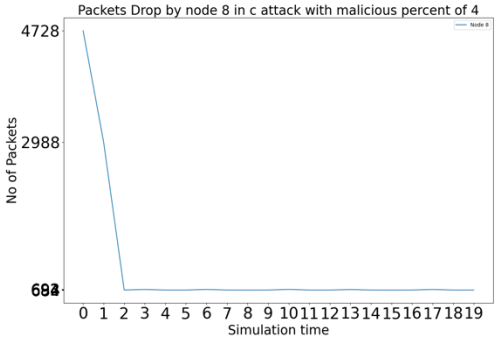

Fig 4.1.4.C.D.8: Packets Dropped by Node 8 with 4 attackers in Constant attack.

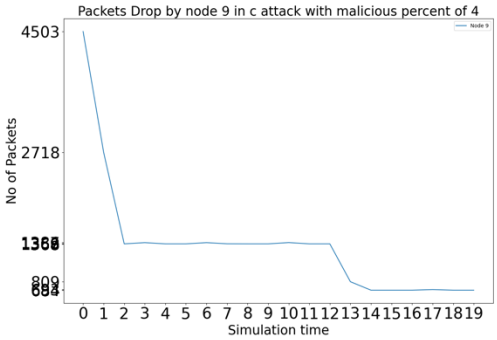

Fig 4.1.4.C.D.9: Packets Dropped by Node 9 with 4 attackers in Constant attack.

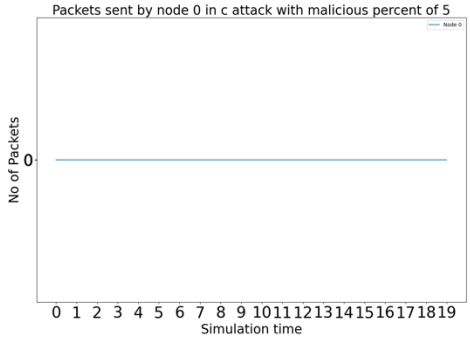

Fig 4.1.5.C.S.0: Packets Sent by Node 0 with 5 attackers in Constant attack.

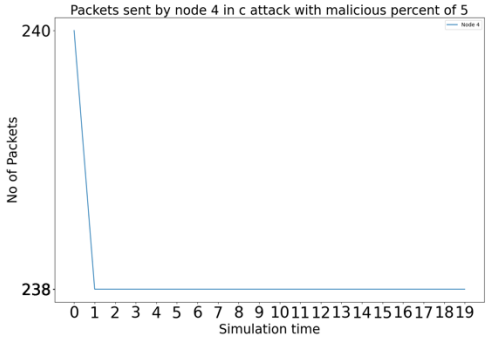

Fig 4.1.5.C.S.4: Packets Sent by Node 4 with 5 attackers in Constant attack.

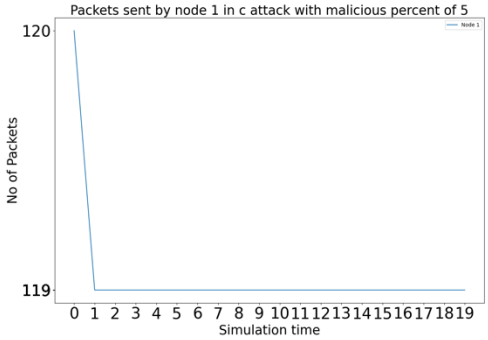

Fig 4.1.5.C.S.1: Packets Sent by Node 1 with 5 attackers in Constant attack.

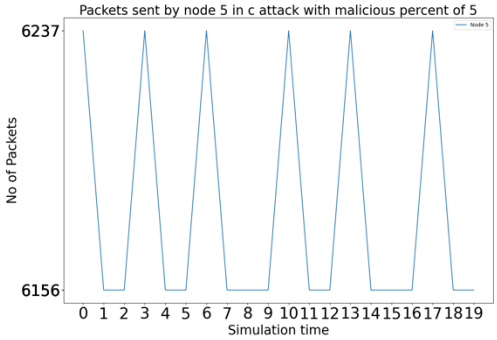

Fig 4.1.5.C.S.5: Packets Sent by Node 5 with 5 attackers in Constant attack.

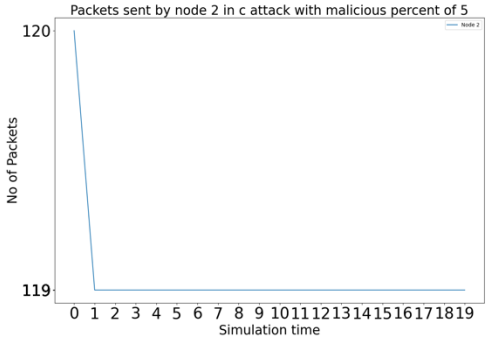

Fig 4.1.5.C.S.2: Packets Sent by Node 2 with 5 attackers in Constant attack.

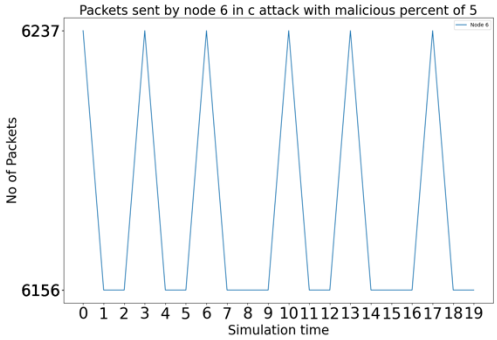

Fig 4.1.5.C.S.6: Packets Sent by Node 6 with 5 attackers in Constant attack.

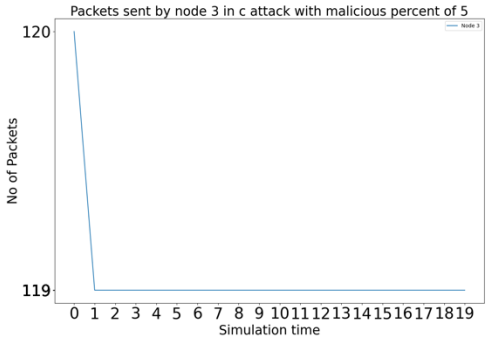

Fig 4.1.5.C.S.3: Packets Sent by Node 3 with 5 attackers in Constant attack.

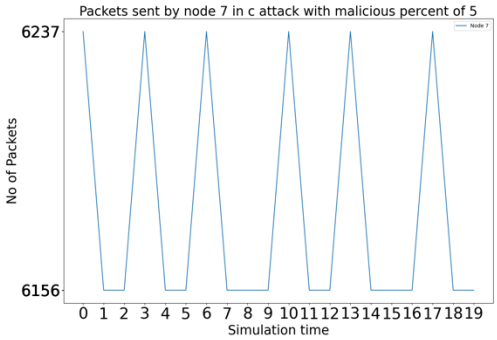

Fig 4.1.5.C.S.7: Packets Sent by Node 7 with 5 attackers in Constant attack.

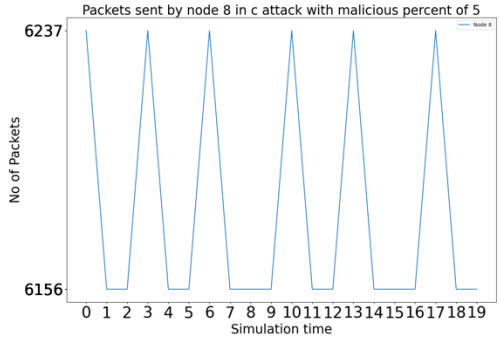

Fig 4.1.5.C.S.8: Packets Sent by Node 8 with 5 attackers in Constant attack.

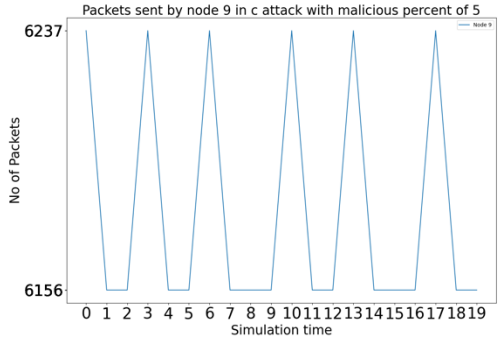

Fig 4.1.5.C.S.9: Packets Sent by Node 9 with 5 attackers in Constant attack.

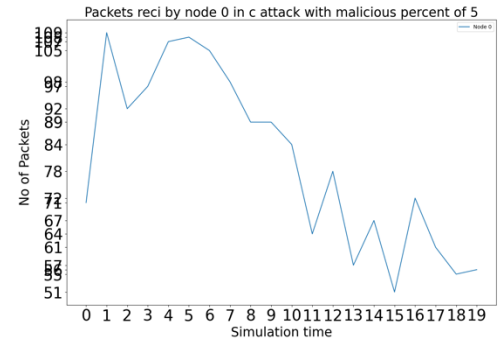

Fig 4.1.5.C.R.0: Packets Received by Node 0 with 5 attackers in Constant attack.

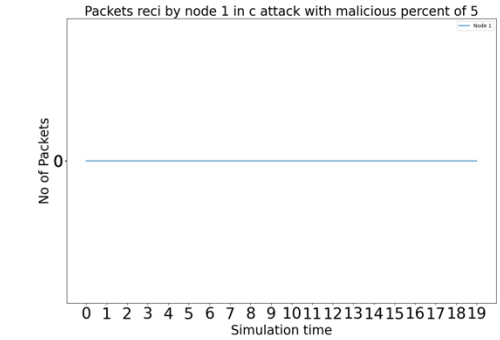

Fig 4.1.5.C.R.1: Packets Received by Node 1 with 5 attackers in Constant attack.

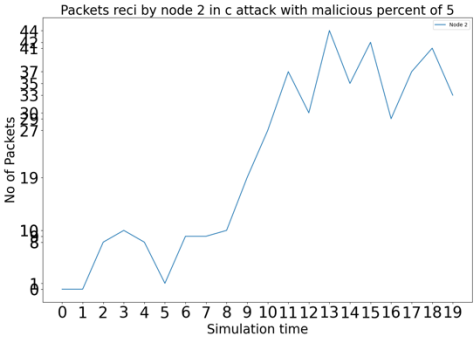

Fig 4.1.5.C.R.2: Packets Received by Node 2 with 5 attackers in Constant attack.

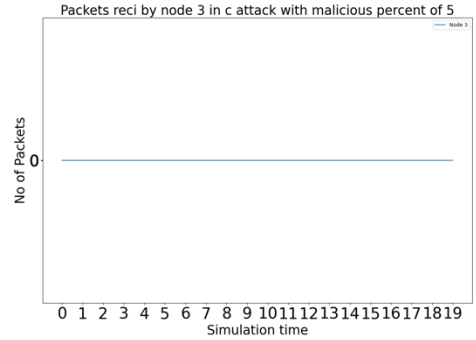

Fig 4.1.5.C.R.3: Packets Received by Node 3 with 5 attackers in Constant attack.

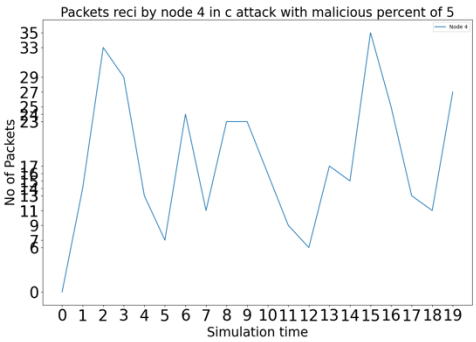

Fig 4.1.5.C.R.4: Packets Received by Node 4 with 5 attackers in Constant attack.

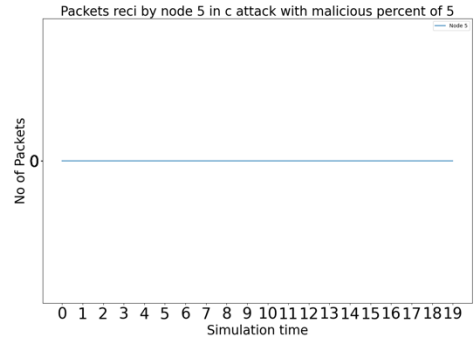

Fig 4.1.5.C.R.5: Packets Received by Node 5 with 5 attackers in Constant attack.

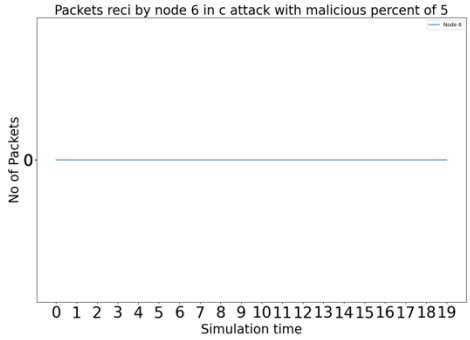

Fig 4.1.5.C.R.6: Packets Received by Node 6 with 5 attackers in Constant attack.

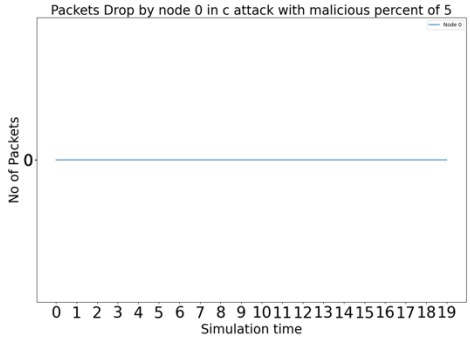

Fig 4.1.5.C.D.0: Packets Dropped by Node 0 with 5 attackers in Constant attack.

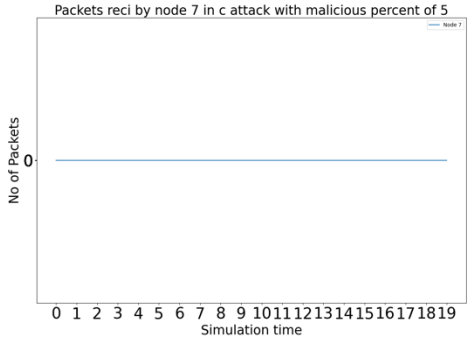

Fig 4.1.5.C.R.7: Packets Received by Node 7 with 5 attackers in Constant attack.

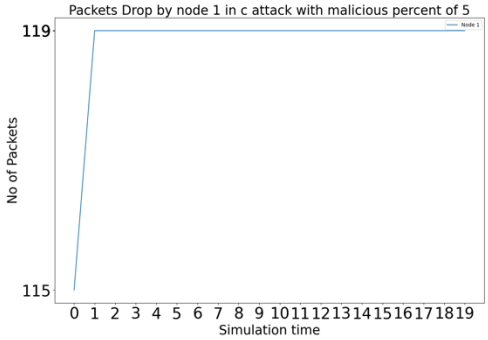

Fig 4.1.5.C.D.1: Packets Dropped by Node 1 with 5 attackers in Constant attack.

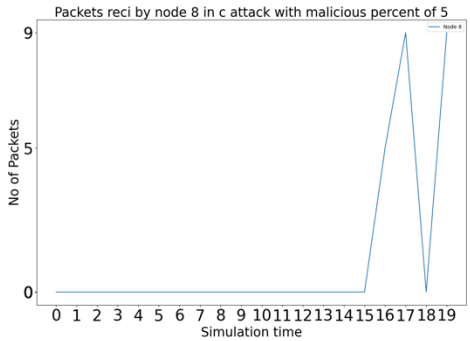

Fig 4.1.5.C.R.8: Packets Received by Node 8 with 5 attackers in Constant attack.

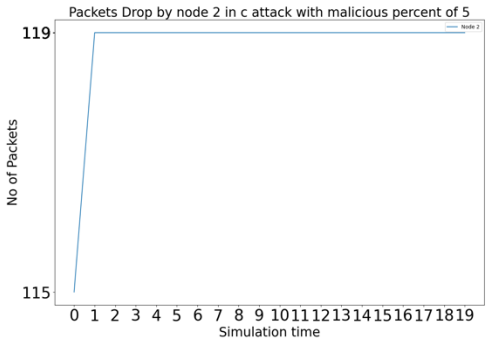

Fig 4.1.5.C.D.2: Packets Dropped by Node 2 with 5 attackers in Constant attack.

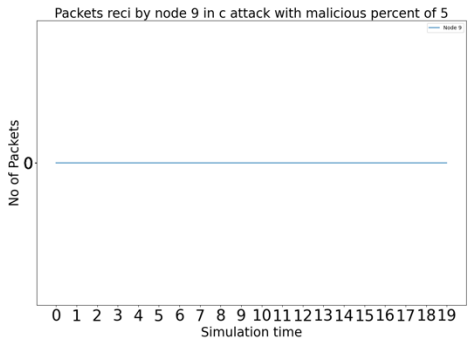

Fig 4.1.5.C.R.9: Packets Received by Node 9 with 5 attackers in Constant attack.

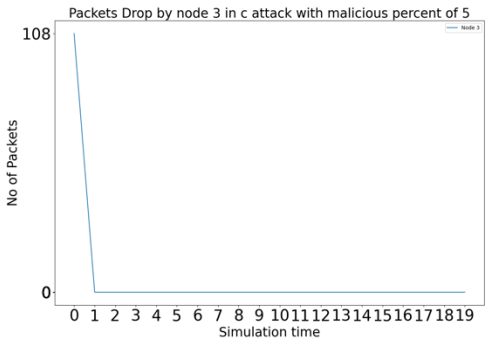

Fig 4.1.5.C.D.3: Packets Dropped by Node 3 with 5 attackers in Constant attack.

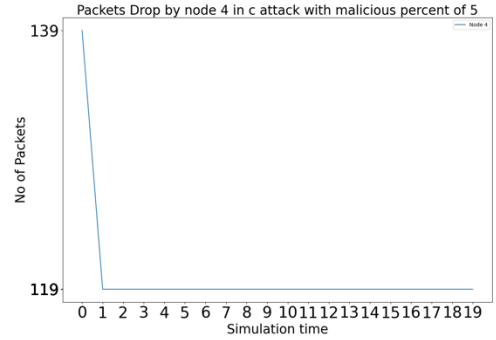

Fig 4.1.5.C.D.4: Packets Dropped by Node 4 with 5 attackers in Constant attack.

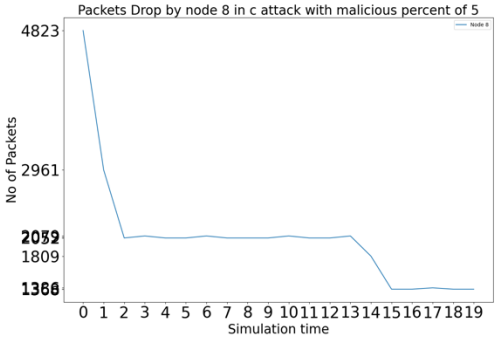

Fig 4.1.5.C.D.8: Packets Dropped by Node 8 with 5 attackers in Constant attack.

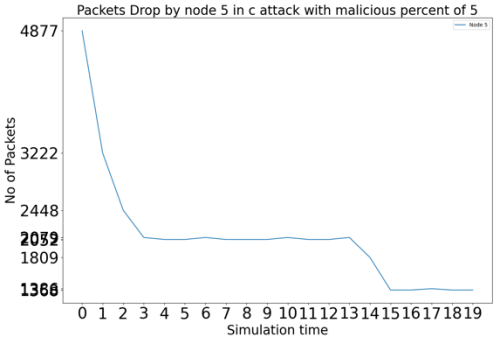

Fig 4.1.5.C.D.5: Packets Dropped by Node 5 with 5 attackers in Constant attack.

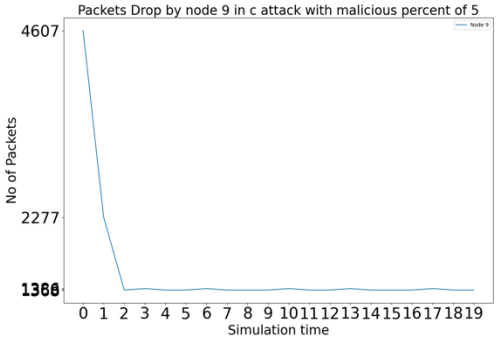

Fig 4.1.5.C.D.9: Packets Dropped by Node 9 with 5 attackers in Constant attack.

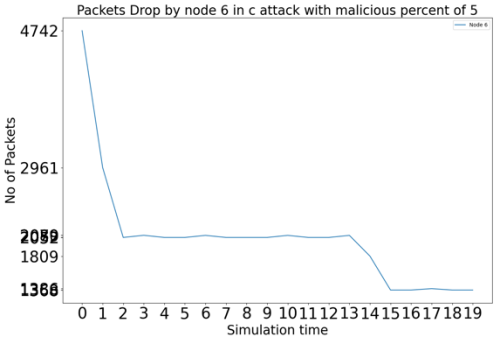

Fig 4.1.5.C.D.6: Packets Dropped by Node 6 with 5 attackers in Constant attack.

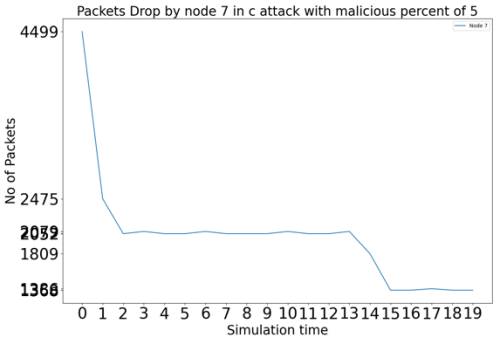

Fig 4.1.5.C.D.7: Packets Dropped by Node 7 with 5 attackers in Constant attack.

Periodic Attack:

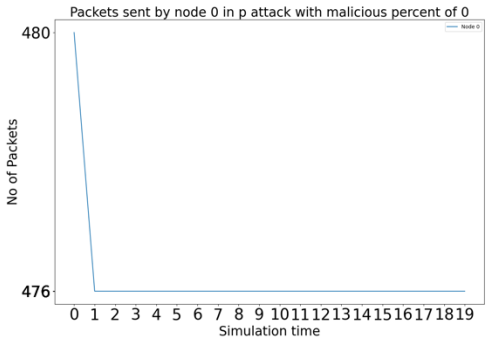

Fig 4.1.0.P.S.0: Packets Sent by Node 0 with 0 attackers in Periodic attack.

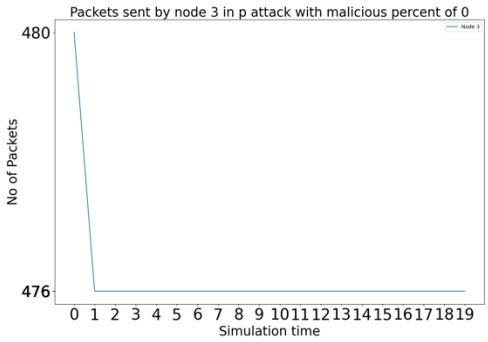

Fig 4.1.0.P.S.3: Packets Sent by Node 3 with 0 attackers in Periodic attack.

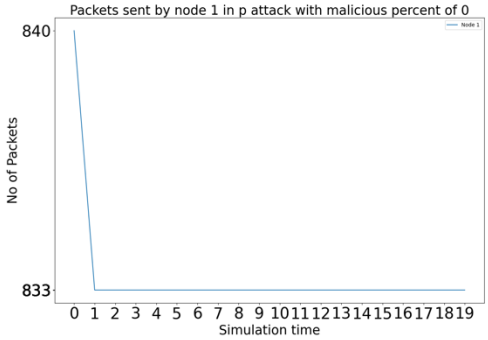

Fig 4.1.0.P.S.1: Packets Sent by Node 1 with 0 attackers in Periodic attack.

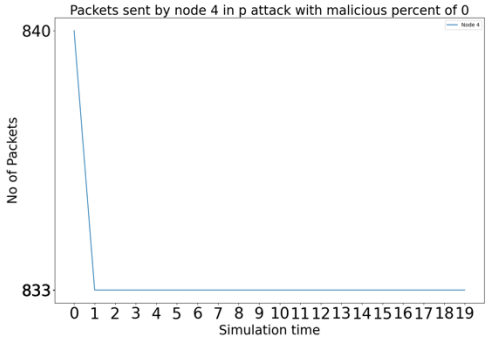

Fig 4.1.0.P.S.4: Packets Sent by Node 4 with 0 attackers in Periodic attack.

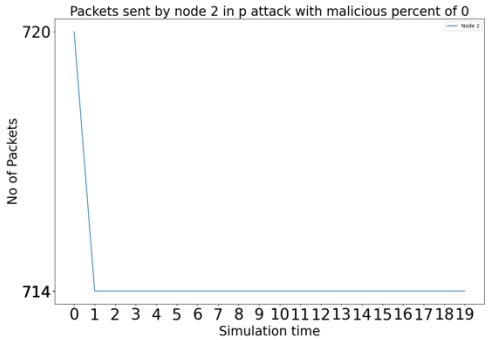

Fig 4.1.0.P.S.2: Packets Sent by Node 2 with 0 attackers in Periodic attack.

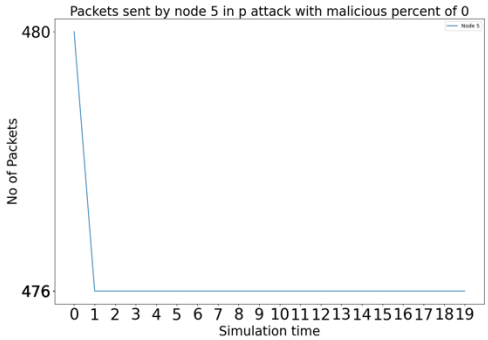

Fig 4.1.0.P.S.5: Packets Sent by Node 5 with 0 attackers in Periodic attack.

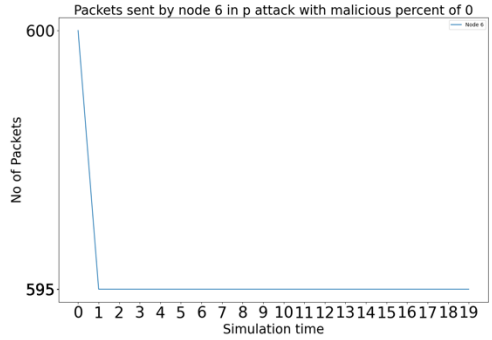

Fig 4.1.0.P.S.6: Packets Sent by Node 6 with 0 attackers in Periodic attack.

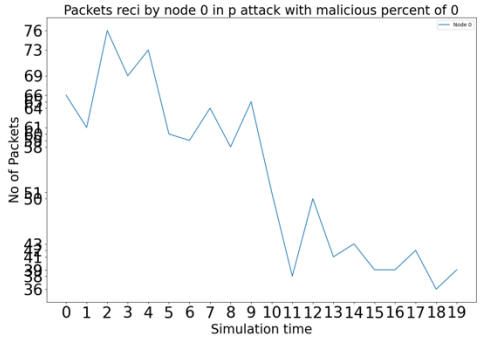

Fig 4.1.0.P.R.0: Packets Received by Node 0 with 0 attackers in Periodic attack.

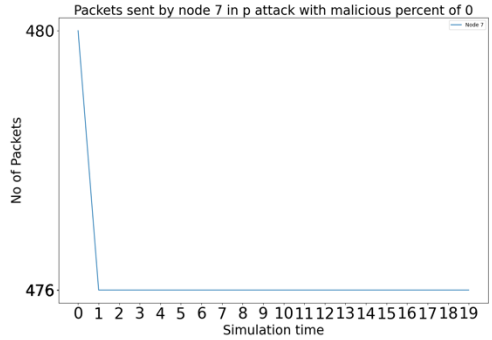

Fig 4.1.0.P.S.7: Packets Sent by Node 7 with 0 attackers in Periodic attack.

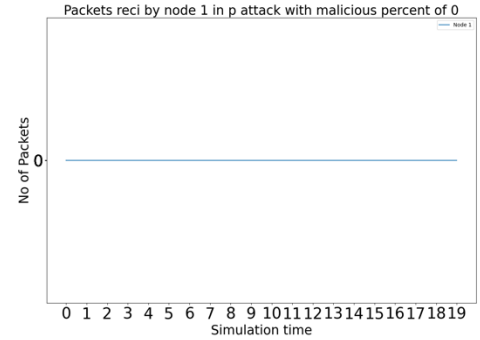

Fig 4.1.0.P.R.1: Packets Received by Node 1 with 0 attackers in Periodic attack.

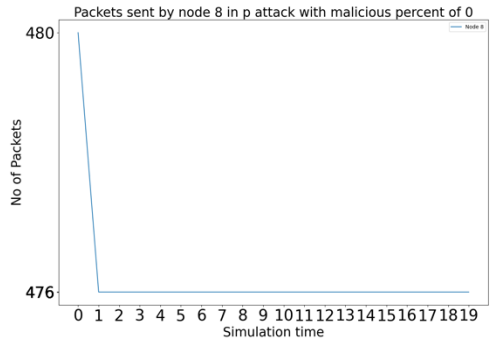

Fig 4.1.0.P.S.8: Packets Sent by Node 8 with 0 attackers in Periodic attack.

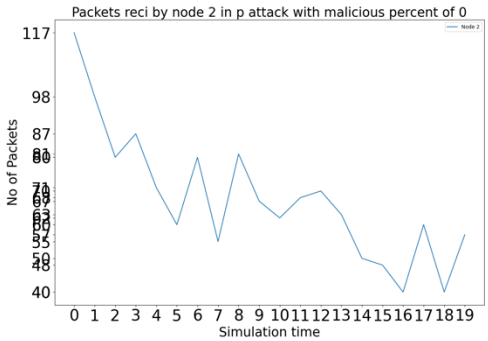

Fig 4.1.0.P.R.2: Packets Received by Node 2 with 0 attackers in Periodic attack.

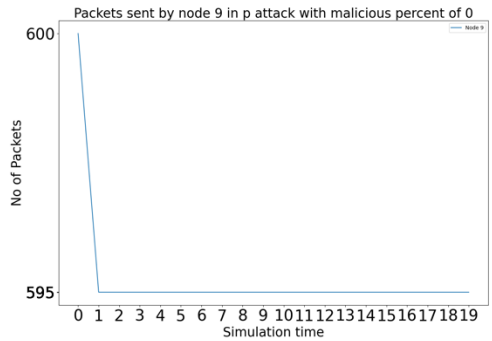

Fig 4.1.0.P.S.9: Packets Sent by Node 9 with 0 attackers in Periodic attack.

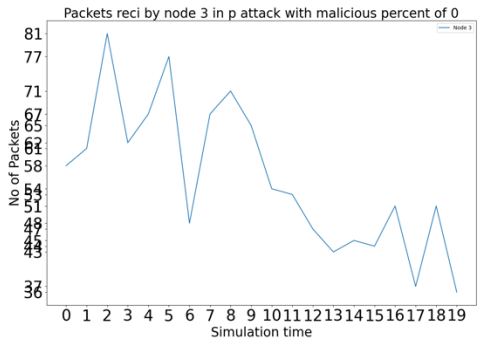

Fig 4.1.0.P.R.3: Packets Received by Node 3 with 0 attackers in Periodic attack.

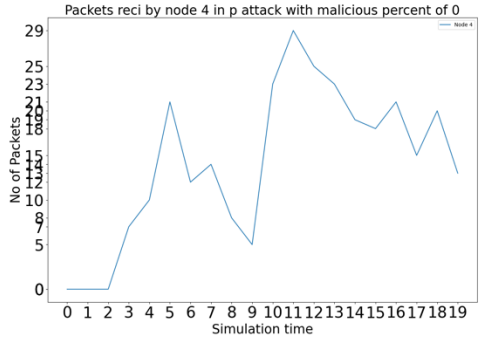

Fig 4.1.0.P.R.4: Packets Received by Node 4 with 0 attackers in Periodic attack.

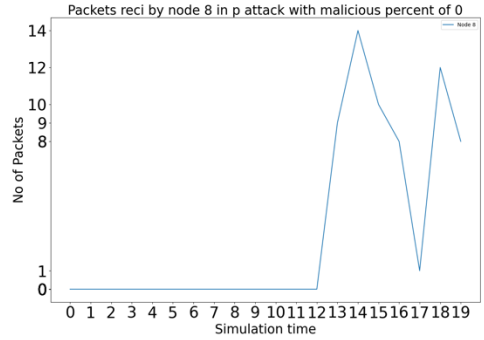

Fig 4.1.0.P.R.8: Packets Received by Node 8 with 0 attackers in Periodic attack.

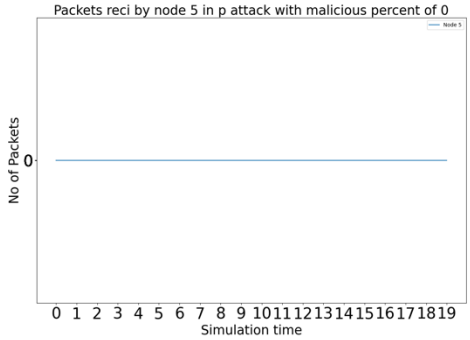

Fig 4.1.0.P.R.5: Packets Received by Node 5 with 0 attackers in Periodic attack.

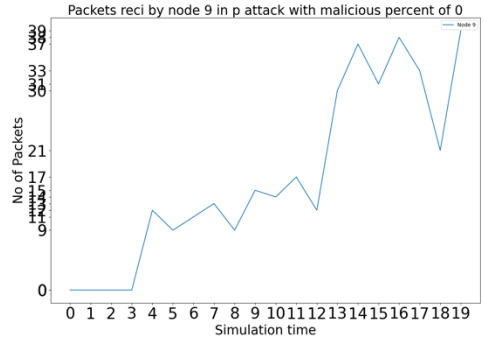

Fig 4.1.0.P.R.9: Packets Received by Node 9 with 0 attackers in Periodic attack.

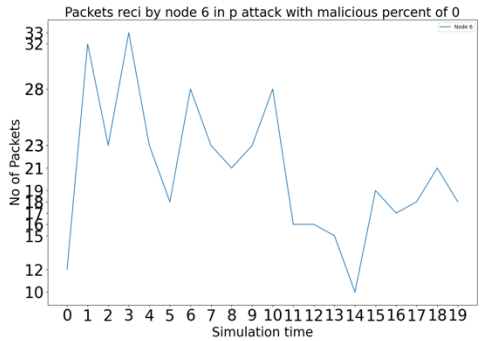

Fig 4.1.0.P.R.6: Packets Received by Node 6 with 0 attackers in Periodic attack.

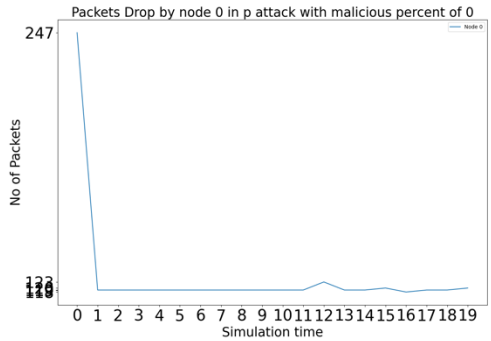

Fig 4.1.0.P.D.0: Packets Dropped by Node 0 with 0 attackers in Periodic attack.

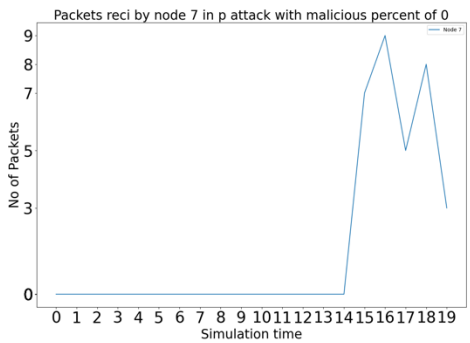

Fig 4.1.0.P.R.7: Packets Received by Node 7 with 0 attackers in Periodic attack.

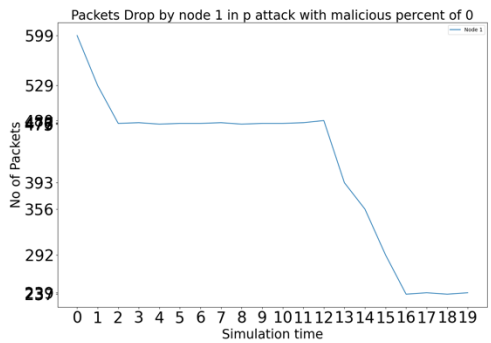

Fig 4.1.0.P.D.1: Packets Dropped by Node 1 with 0 attackers in Periodic attack.

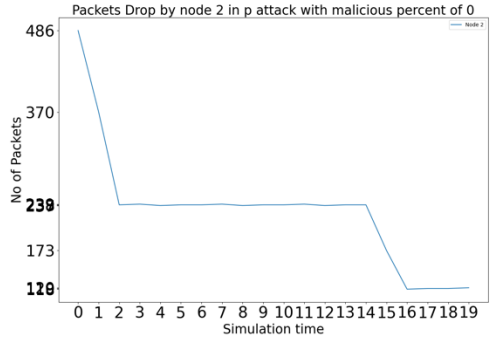

Fig 4.1.0.P.D.2: Packets Dropped by Node 2 with 0 attackers in Periodic attack.

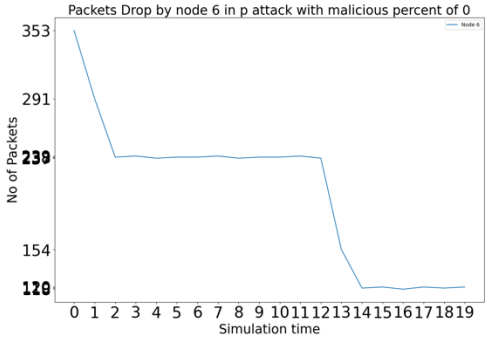

Fig 4.1.0.P.D.6: Packets Dropped by Node 6 with 0 attackers in Periodic attack.

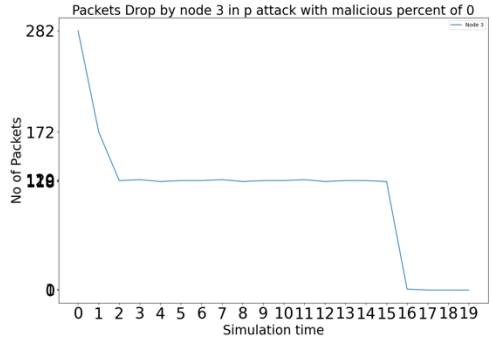

Fig 4.1.0.P.D.3: Packets Dropped by Node 3 with 0 attackers in Periodic attack.

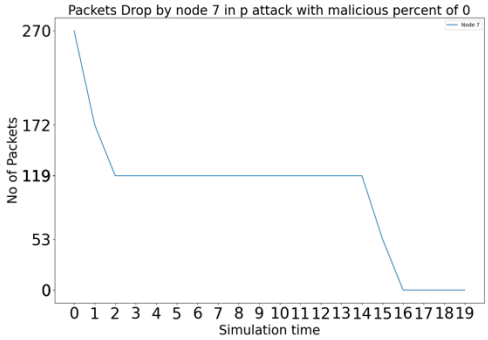

Fig 4.1.0.P.D.7: Packets Dropped by Node 7 with 0 attackers in Periodic attack.

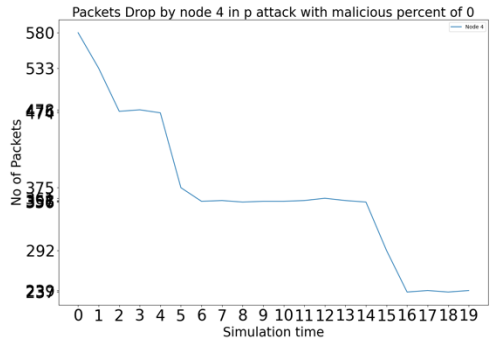

Fig 4.1.0.P.D.4: Packets Dropped by Node 4 with 0 attackers in Periodic attack.

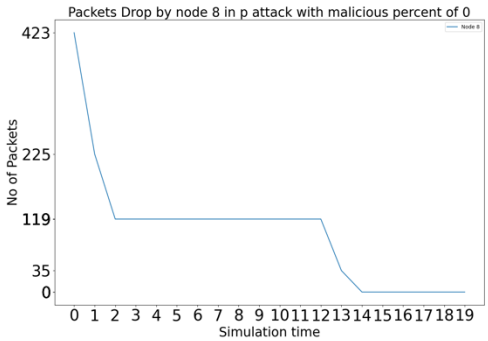

Fig 4.1.0.P.D.8: Packets Dropped by Node 8 with 0 attackers in Periodic attack.

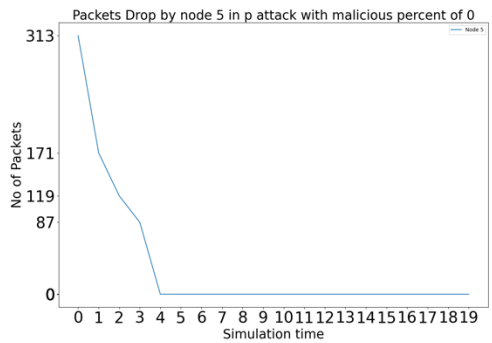

Fig 4.1.0.P.D.5: Packets Dropped by Node 5 with 0 attackers in Periodic attack.

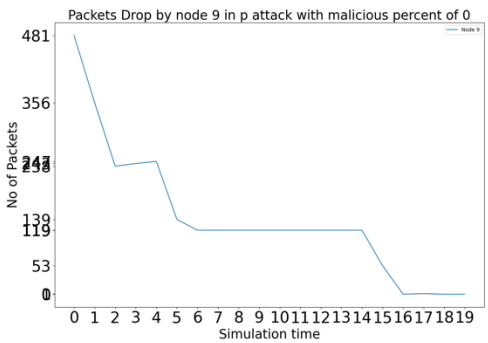

Fig 4.1.0.P.D.9: Packets Dropped by Node 9 with 0 attackers in Periodic attack.

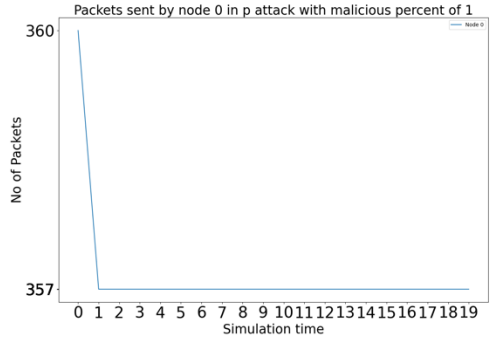

Fig 4.1.1.P.S.0: Packets Sent by Node 0 with 1 attacker in Periodic attack.

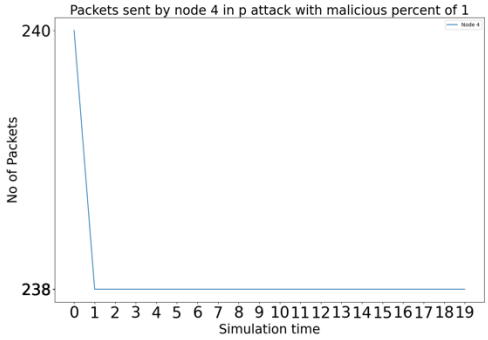

Fig 4.1.1.P.S.4: Packets Sent by Node 4 with 1 attacker in Periodic attack.

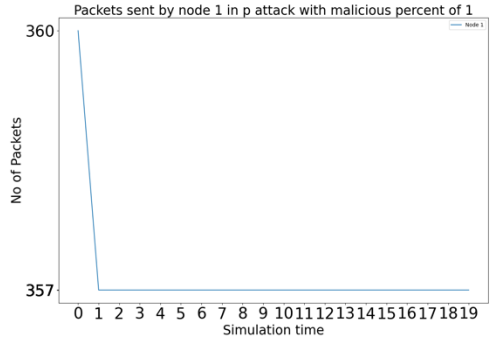

Fig 4.1.1.P.S.1: Packets Sent by Node 1 with 1 attacker in Periodic attack.

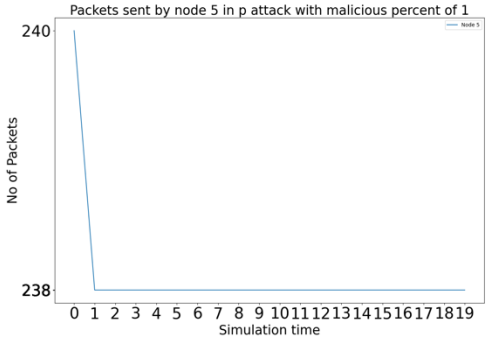

Fig 4.1.1.P.S.5: Packets Sent by Node 5 with 1 attacker in Periodic attack.

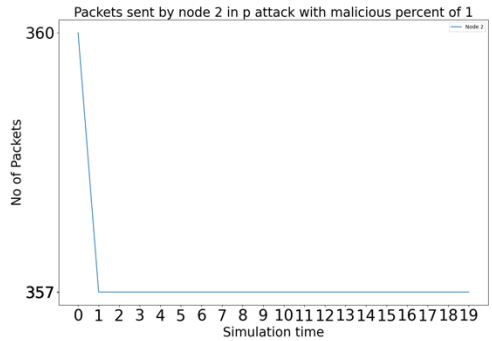

Fig 4.1.1.P.S.2: Packets Sent by Node 2 with 1 attacker in Periodic attack.

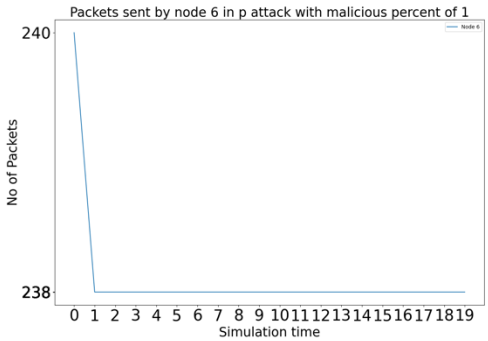

Fig 4.1.1.P.S.6: Packets Sent by Node 6 with 1 attacker in Periodic attack.

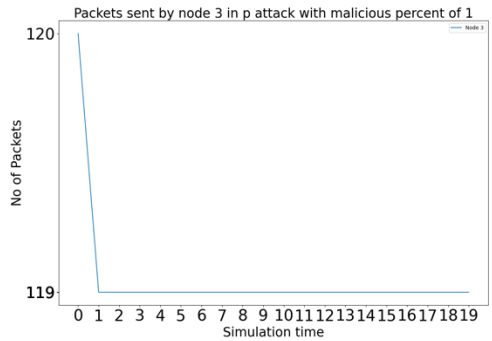

Fig 4.1.1.P.S.3: Packets Sent by Node 3 with 1 attacker in Periodic attack.

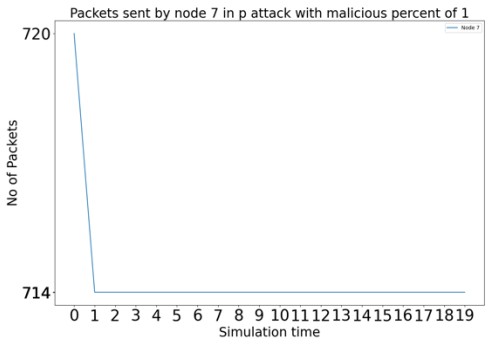

Fig 4.1.1.P.S.7: Packets Sent by Node 7 with 1 attacker in Periodic attack.

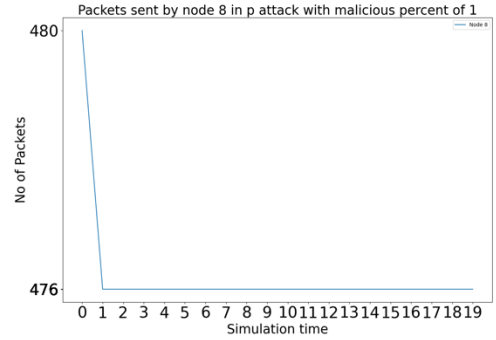

Fig 4.1.1.P.S.8: Packets Sent by Node 8 with 1 attacker in Periodic attack.

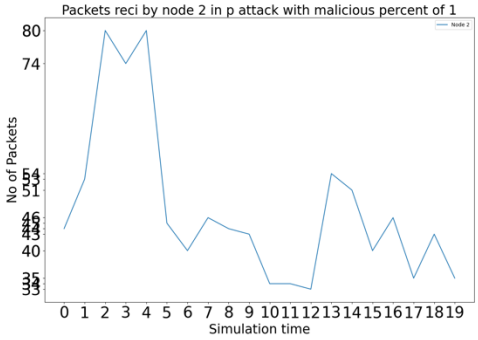

Fig 4.1.1.P.R.2: Packets Received by Node 2 with 1 attacker in Periodic attack.

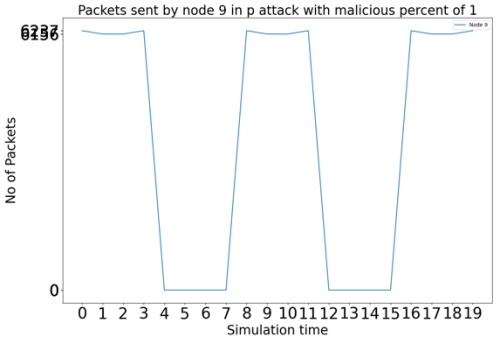

Fig 4.1.1.P.S.9: Packets Sent by Node 9 with 1 attacker in Periodic attack.

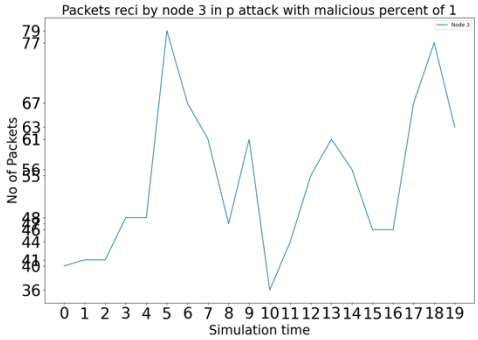

Fig 4.1.1.P.R.3: Packets Received by Node 3 with 1 attacker in Periodic attack.

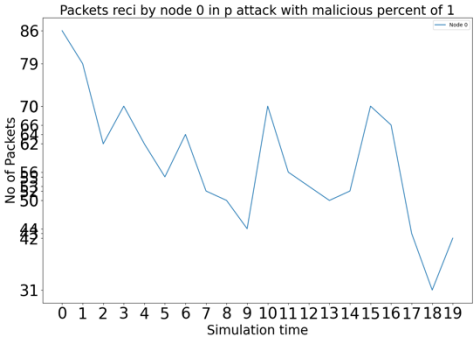

Fig 4.1.1.P.R.0: Packets Received by Node 0 with 1 attacker in Periodic attack.

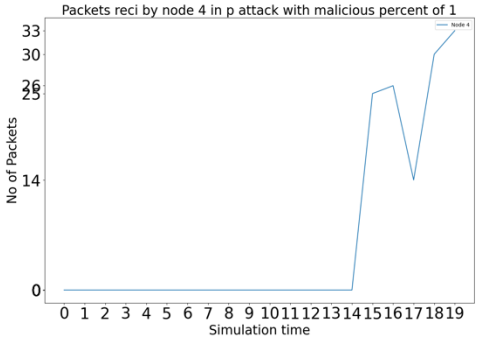

Fig 4.1.1.P.R.4: Packets Received by Node 4 with 1 attacker in Periodic attack.

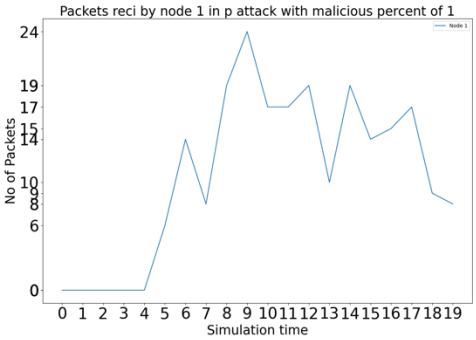

Fig 4.1.1.P.R.1: Packets Received by Node 1 with 1 attacker in Periodic attack.

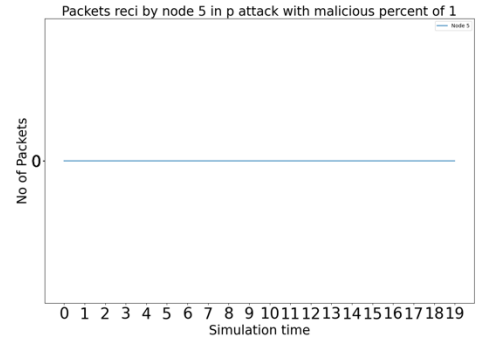

Fig 4.1.1.P.R.5: Packets Received by Node 5 with 1 attacker in Periodic attack.

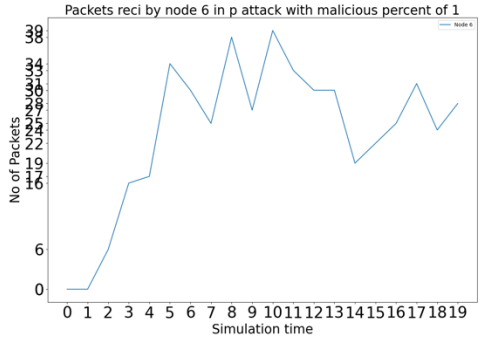

Fig 4.1.1.P.R.6: Packets Received by Node 6 with 1 attacker in Periodic attack.

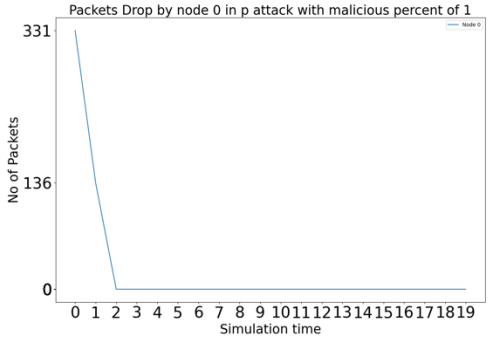

Fig 4.1.1.P.D.0: Packets Dropped by Node 0 with 1 attacker in Periodic attack.

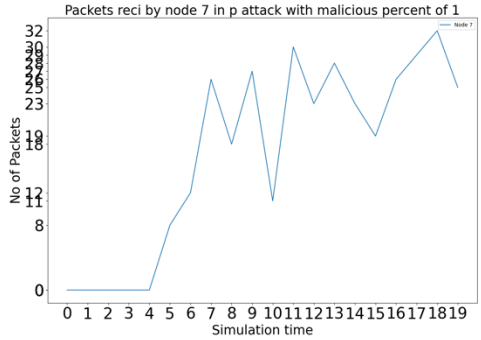

Fig 4.1.1.P.R.7: Packets Received by Node 7 with 1 attacker in Periodic attack.

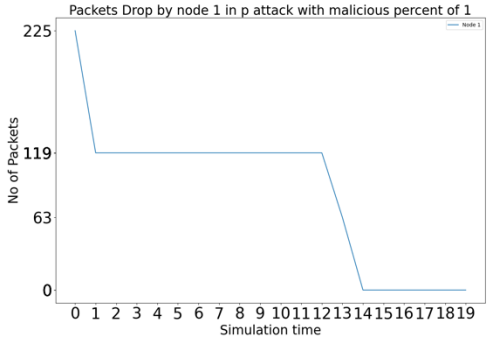

Fig 4.1.1.P.D.1: Packets Dropped by Node 1 with 1 attacker in Periodic attack.

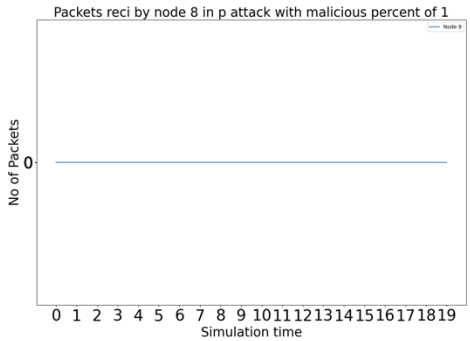

Fig 4.1.1.P.R.8: Packets Received by Node 8 with 1 attacker in Periodic attack.

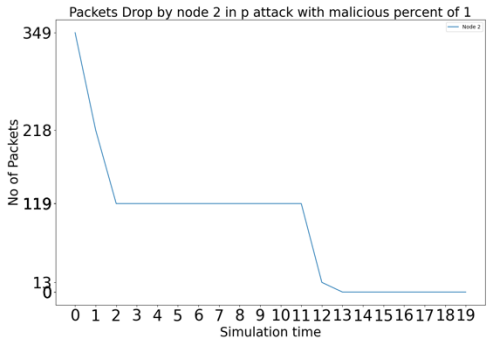

Fig 4.1.1.P.D.2: Packets Dropped by Node 2 with 1 attacker in Periodic attack.

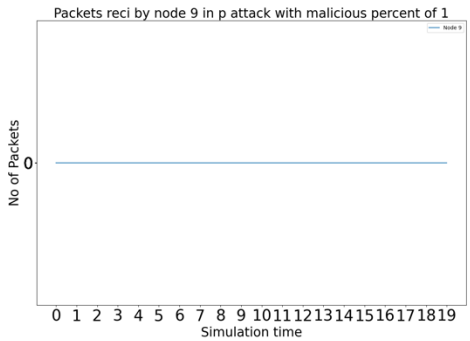

Fig 4.1.1.P.R.9: Packets Received by Node 9 with 1 attacker in Periodic attack.

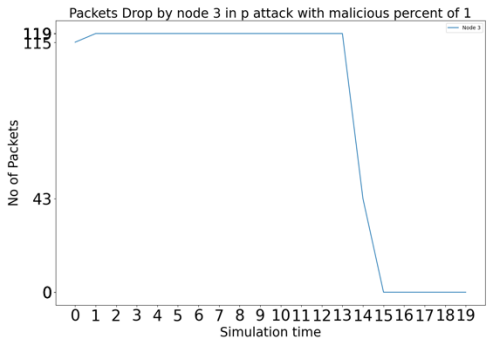

Fig 4.1.1.P.D.3: Packets Dropped by Node 3 with 1 attacker in Periodic attack.

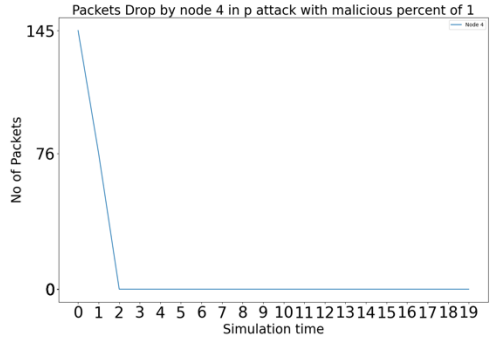

Fig 4.1.1.P.D.4: Packets Dropped by Node 4 with 1 attacker in Periodic attack.

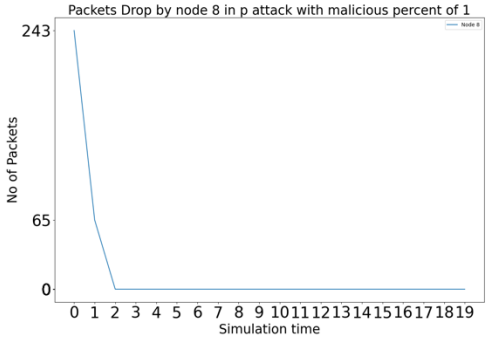

Fig 4.1.1.P.D.8: Packets Dropped by Node 8 with 1 attacker in Periodic attack.

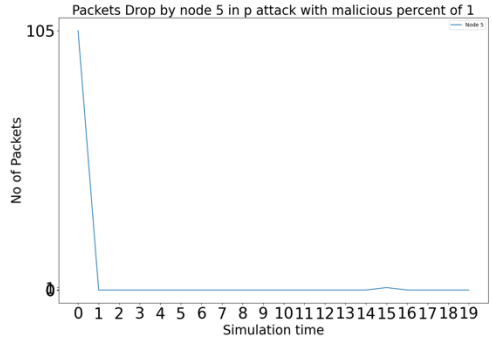

Fig 4.1.1.P.D.5: Packets Dropped by Node 5 with 1 attacker in Periodic attack.

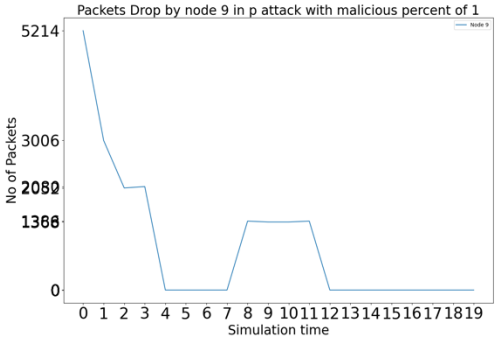

Fig 4.1.1.P.D.9: Packets Dropped by Node 9 with 1 attacker in Periodic attack.

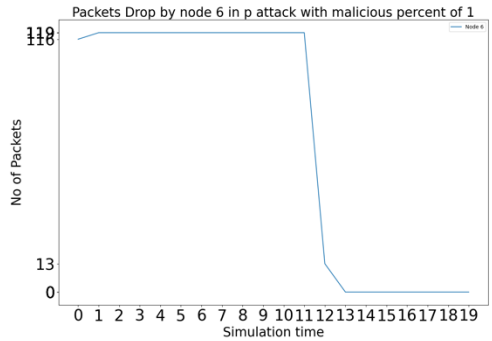

Fig 4.1.1.P.D.6: Packets Dropped by Node 6 with 1 attacker in Periodic attack.

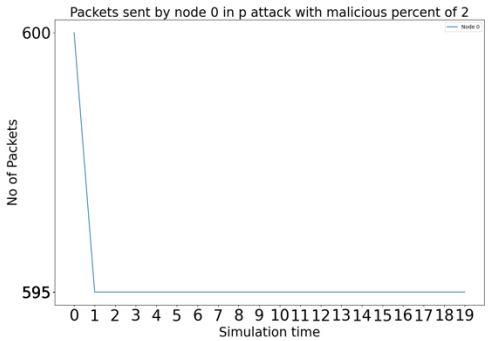

Fig 4.1.2.P.S.0: Packets Sent by Node 0 with 2 attackers in Periodic attack.

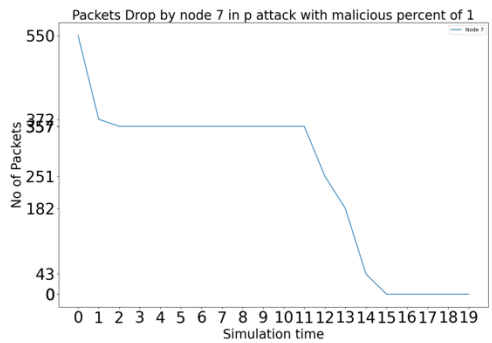

Fig 4.1.1.P.D.7: Packets Dropped by Node 7 with 1 attacker in Periodic attack.

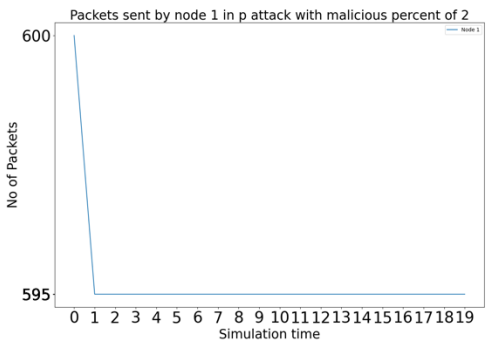

Fig 4.1.2.P.S.1: Packets Sent by Node 1 with 2 attackers in Periodic attack.

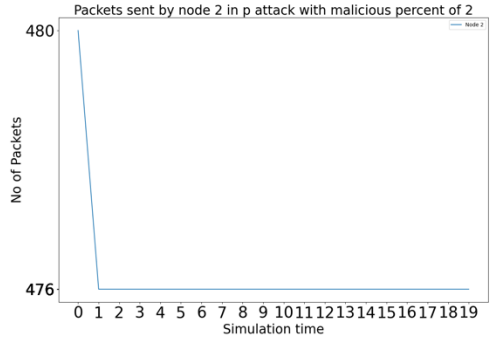

Fig 4.1.2.P.S.2: Packets Sent by Node 2 with 2 attackers in Periodic attack.

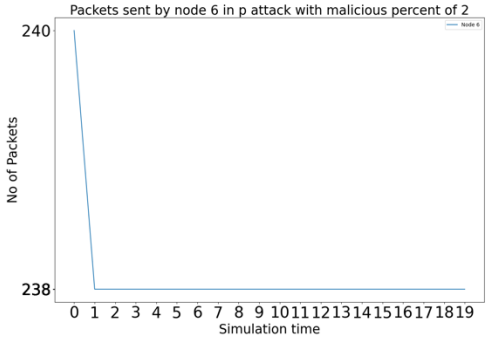

Fig 4.1.2.P.S.6: Packets Sent by Node 6 with 2 attackers in Periodic attack.

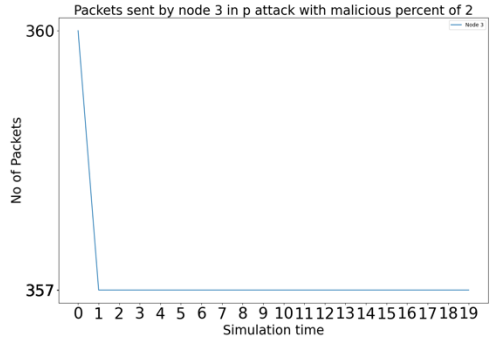

Fig 4.1.2.P.S.3: Packets Sent by Node 3 with 2 attackers in Periodic attack.

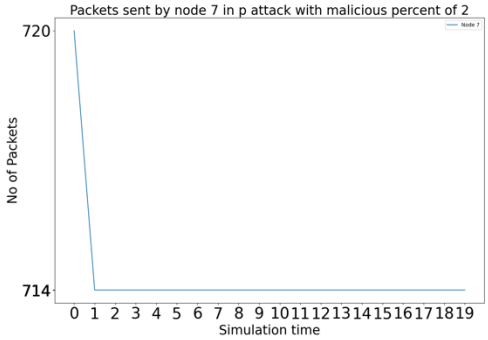

Fig 4.1.2.P.S.7: Packets Sent by Node 7 with 2 attackers in Periodic attack.

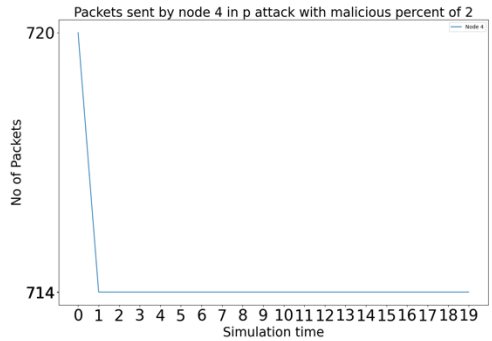

Fig 4.1.2.P.S.4: Packets Sent by Node 4 with 2 attackers in Periodic attack.

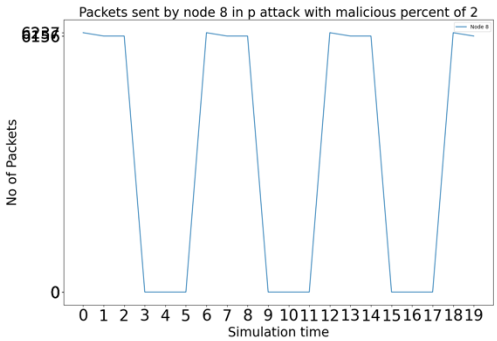

Fig 4.1.2.P.S.8: Packets Sent by Node 8 with 2 attackers in Periodic attack.

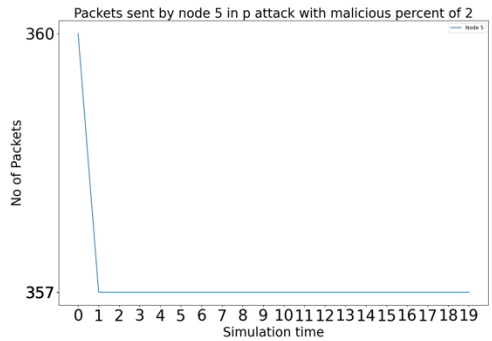

Fig 4.1.2.P.S.5: Packets Sent by Node 5 with 2 attackers in Periodic attack.

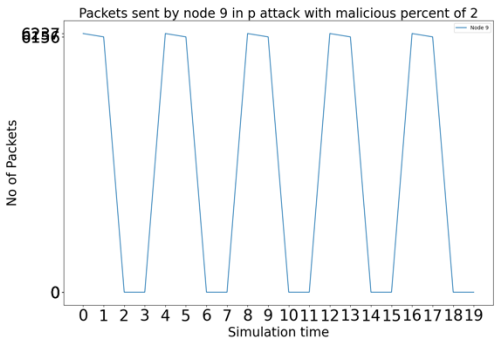

Fig 4.1.2.P.S.9: Packets Sent by Node 9 with 2 attackers in Periodic attack.

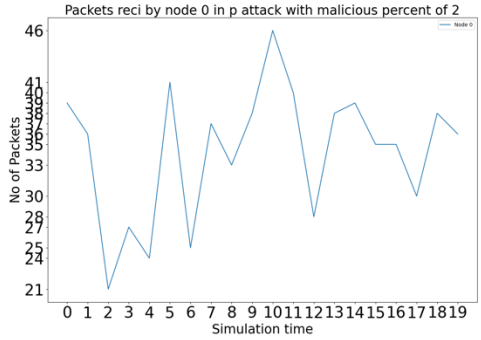

Fig 4.1.2.P.R.0: Packets Received by Node 0 with 2 attackers in Periodic attack.

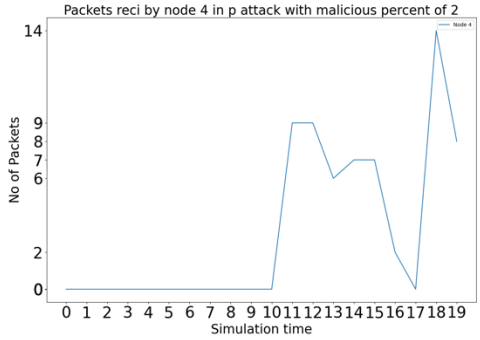

Fig 4.1.2.P.R.4: Packets Received by Node 4 with 2 attackers in Periodic attack.

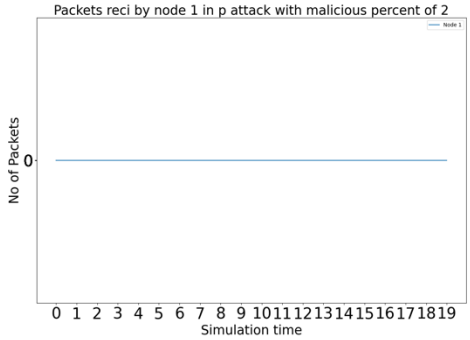

Fig 4.1.2.P.R.1: Packets Received by Node 1 with 2 attackers in Periodic attack.

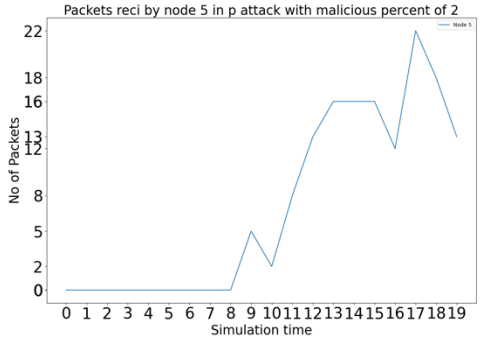

Fig 4.1.2.P.R.5: Packets Received by Node 5 with 2 attackers in Periodic attack.

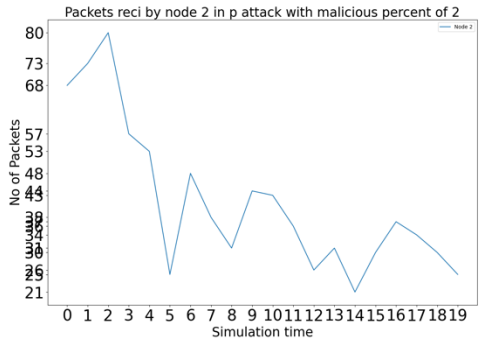

Fig 4.1.2.P.R.2: Packets Received by Node 2 with 2 attackers in Periodic attack.

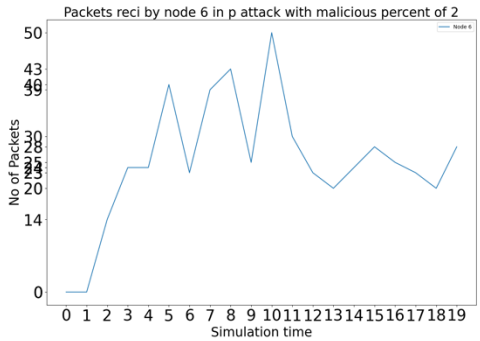

Fig 4.1.2.P.R.6: Packets Received by Node 6 with 2 attackers in Periodic attack.

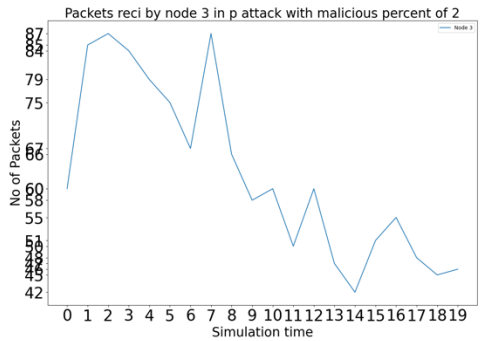

Fig 4.1.2.P.R.3: Packets Received by Node 3 with 2 attackers in Periodic attack.

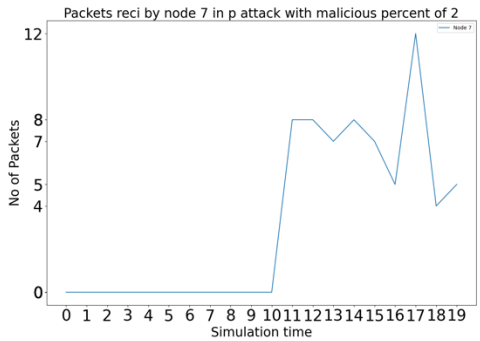

Fig 4.1.2.P.R.7: Packets Received by Node 7 with 2 attackers in Periodic attack.

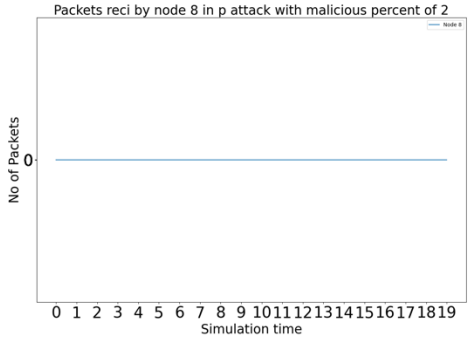

Fig 4.1.2.P.R.8: Packets Received by Node 8 with 2 attackers in Periodic attack.

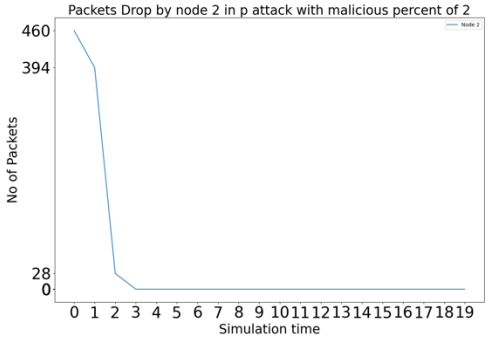

Fig 4.1.2.P.D.2: Packets Dropped by Node 2 with 2 attackers in Periodic attack.

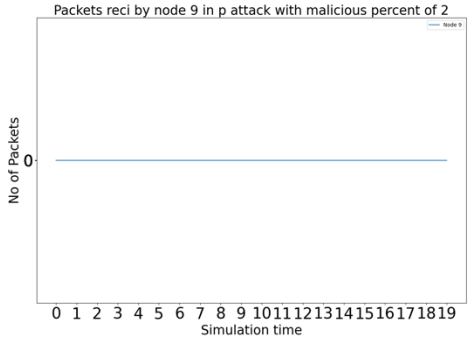

Fig 4.1.2.P.R.9: Packets Received by Node 9 with 2 attackers in Periodic attack.

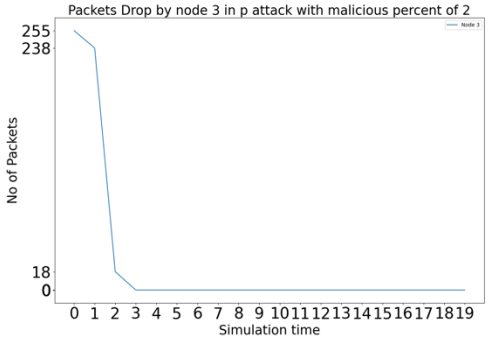

Fig 4.1.2.P.D.3: Packets Dropped by Node 3 with 2 attackers in Periodic attack.

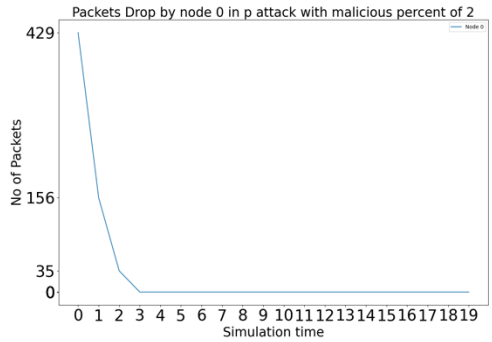

Fig 4.1.2.P.D.0: Packets Dropped by Node 0 with 2 attackers in Periodic attack.

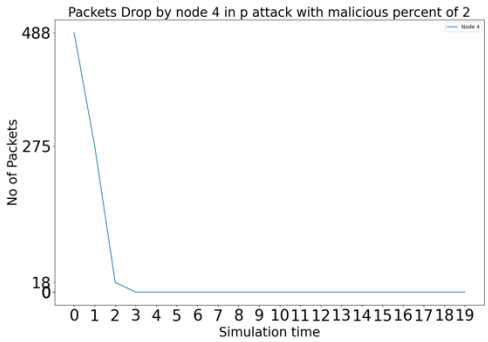

Fig 4.1.2.P.D.4: Packets Dropped by Node 4 with 2 attackers in Periodic attack.

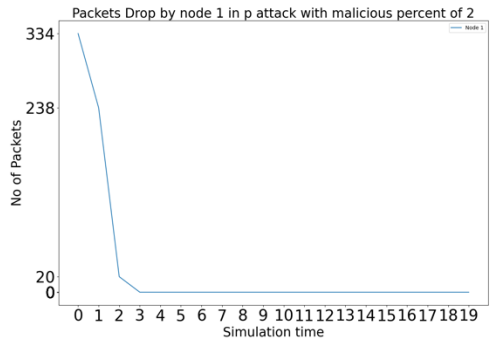

Fig 4.1.2.P.D.1: Packets Dropped by Node 1 with 2 attackers in Periodic attack.

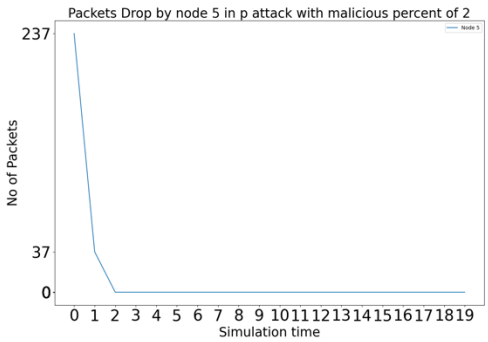

Fig 4.1.2.P.D.5: Packets Dropped by Node 5 with 2 attackers in Periodic attack.

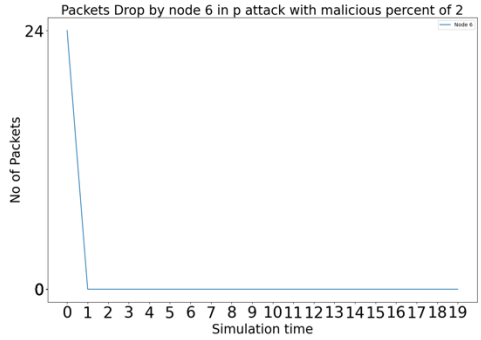

Fig 4.1.2.P.D.6: Packets Dropped by Node 6 with 2 attackers in Periodic attack.

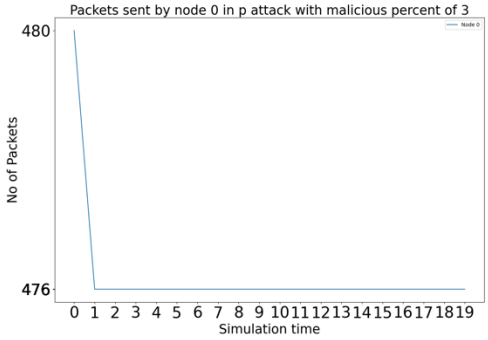

Fig 4.1.3.P.S.0: Packets Sent by Node 0 with 3 attackers in Periodic attack.

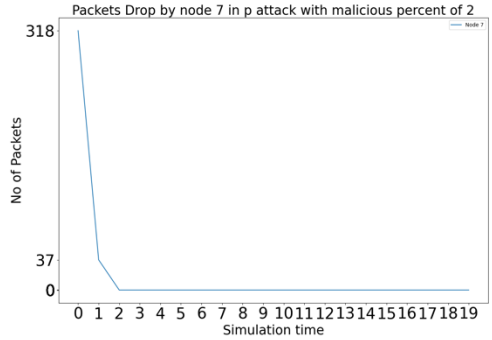

Fig 4.1.2.P.D.7: Packets Dropped by Node 7 with 2 attackers in Periodic attack.

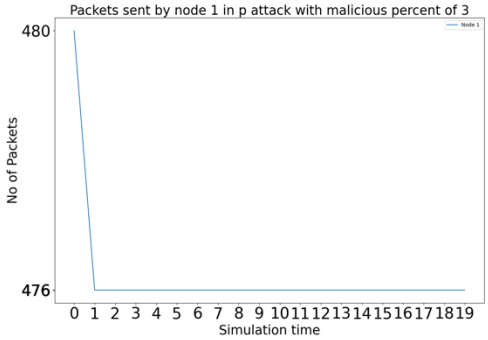

Fig 4.1.3.P.S.1: Packets Sent by Node 1 with 3 attackers in Periodic attack.

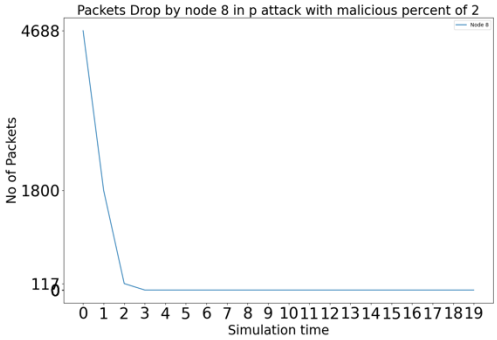

Fig 4.1.2.P.D.8: Packets Dropped by Node 8 with 2 attackers in Periodic attack.

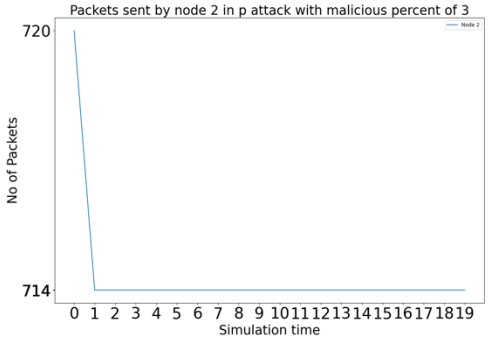

Fig 4.1.3.P.S.2: Packets Sent by Node 2 with 3 attackers in Periodic attack.

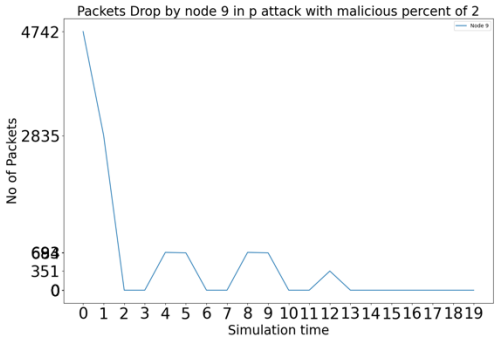

Fig 4.1.2.P.D.9: Packets Dropped by Node 9 with 2 attackers in Periodic attack.

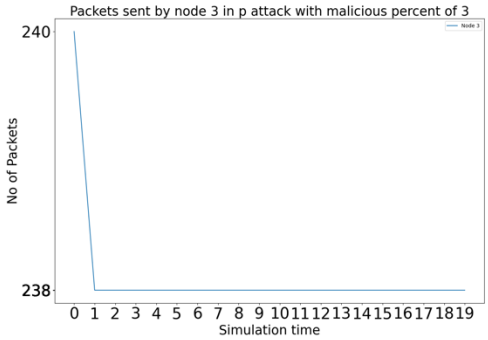

Fig 4.1.3.P.S.3: Packets Sent by Node 3 with 3 attackers in Periodic attack.

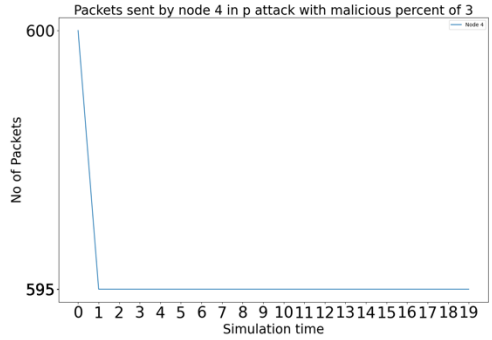

Fig 4.1.3.P.S.4: Packets Sent by Node 4 with 3 attackers in Periodic attack.

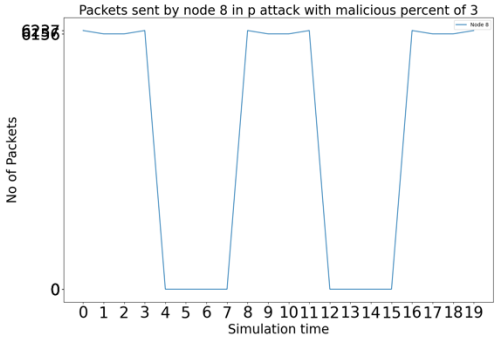

Fig 4.1.3.P.S.8: Packets Sent by Node 8 with 3 attackers in Periodic attack.

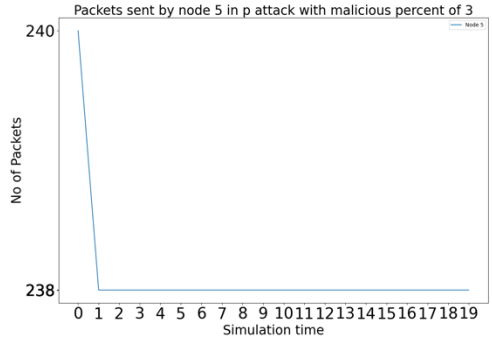

Fig 4.1.3.P.S.5: Packets Sent by Node 5 with 3 attackers in Periodic attack.

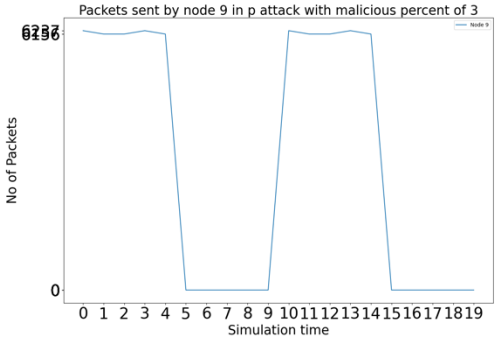

Fig 4.1.3.P.S.9: Packets Sent by Node 9 with 3 attackers in Periodic attack.

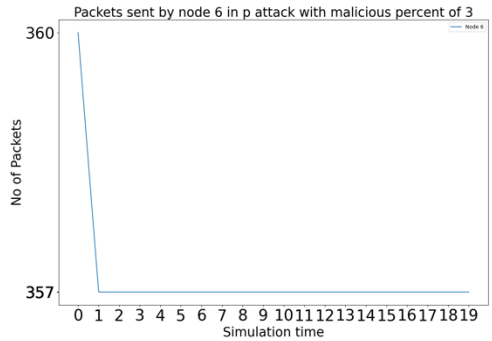

Fig 4.1.3.P.S.6: Packets Sent by Node 6 with 3 attackers in Periodic attack.

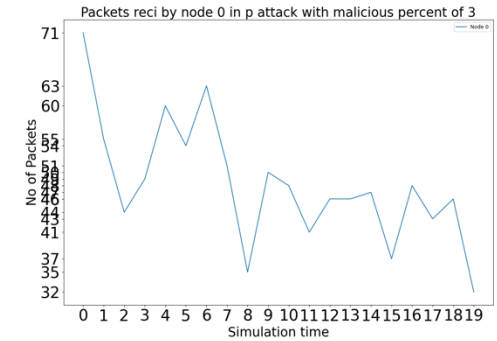

Fig 4.1.3.P.R.0: Packets Received by Node 0 with 3 attackers in Periodic attack.

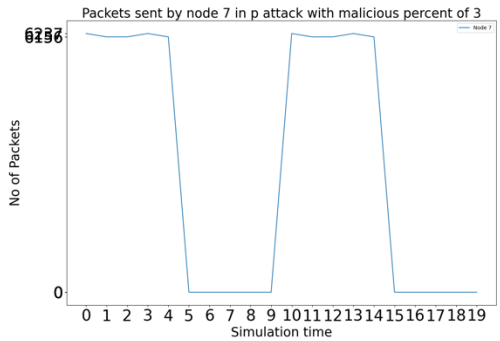

Fig 4.1.3.P.S.7: Packets Sent by Node 7 with 3 attackers in Periodic attack.

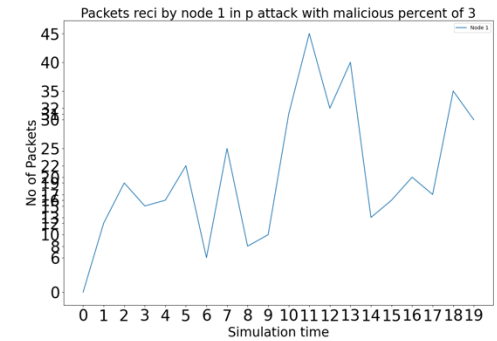

Fig 4.1.3.P.R.1: Packets Received by Node 1 with 3 attackers in Periodic attack.

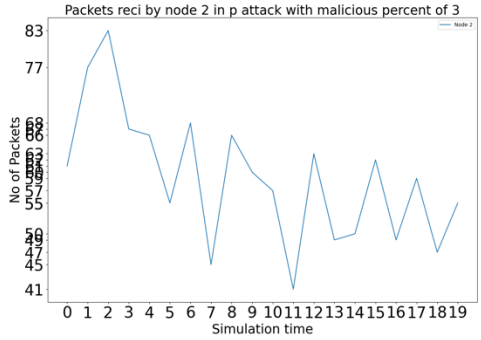

Fig 4.1.3.P.R.2: Packets Received by Node 2 with 3 attackers in Periodic attack.

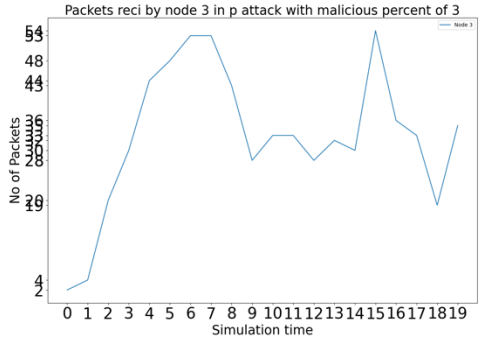

Fig 4.1.3.P.R.3: Packets Received by Node 3 with 3 attackers in Periodic attack.

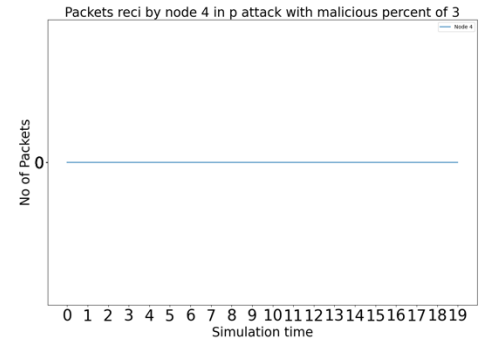

Fig 4.1.3.P.R.4: Packets Received by Node 4 with 3 attackers in Periodic attack.

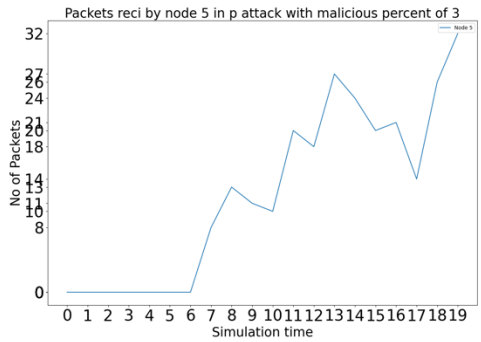

Fig 4.1.3.P.R.5: Packets Received by Node 5 with 3 attackers in Periodic attack.

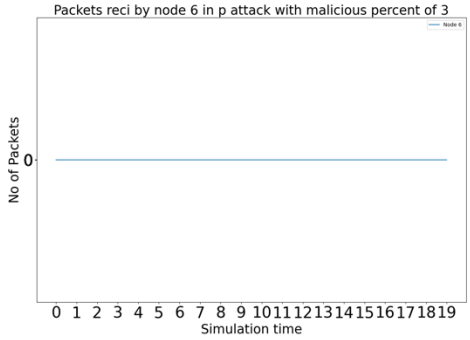

Fig 4.1.3.P.R.6: Packets Received by Node 6 with 3 attackers in Periodic attack.

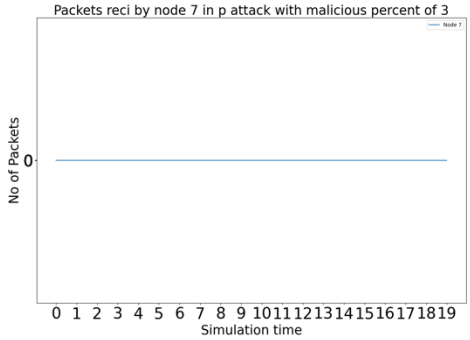

Fig 4.1.3.P.R.7: Packets Received by Node 7 with 3 attackers in Periodic attack.

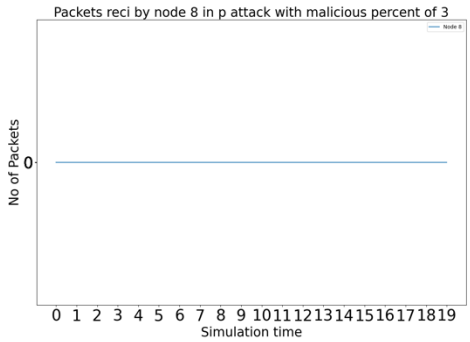

Fig 4.1.3.P.R.8: Packets Received by Node 8 with 3 attackers in Periodic attack.

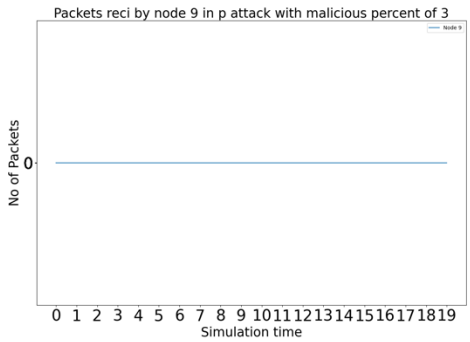

Fig 4.1.3.P.R.9: Packets Received by Node 9 with 3 attackers in Periodic attack.

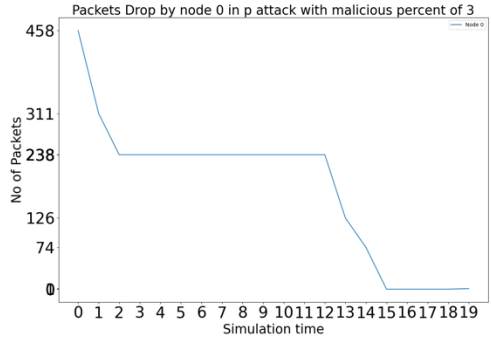

Fig 4.1.3.P.D.0: Packets Dropped by Node 0 with 3 attackers in Periodic attack.

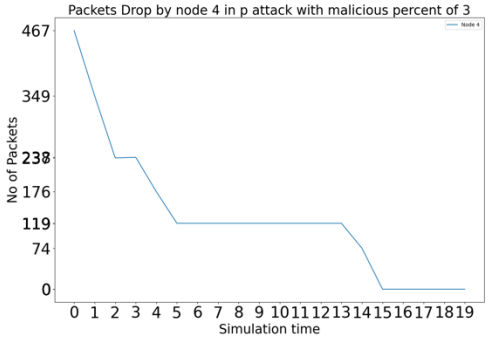

Fig 4.1.3.P.D.4: Packets Dropped by Node 4 with 3 attackers in Periodic attack.

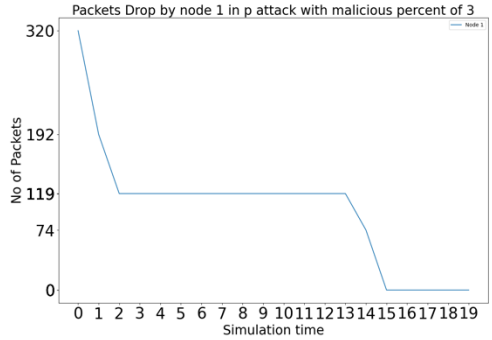

Fig 4.1.3.P.D.1: Packets Dropped by Node 1 with 3 attackers in Periodic attack.

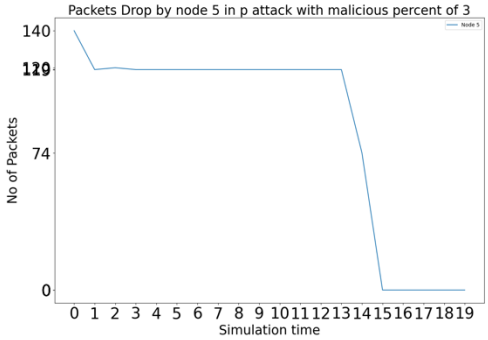

Fig 4.1.3.P.D.5: Packets Dropped by Node 5 with 3 attackers in Periodic attack.

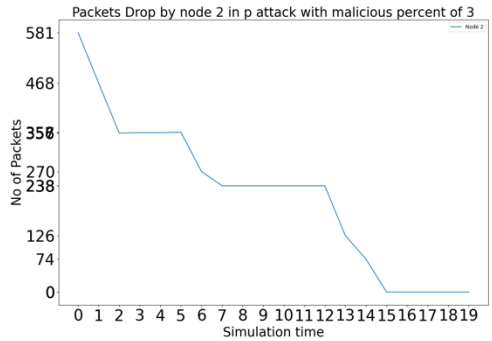

Fig 4.1.3.P.D.2: Packets Dropped by Node 2 with 3 attackers in Periodic attack.

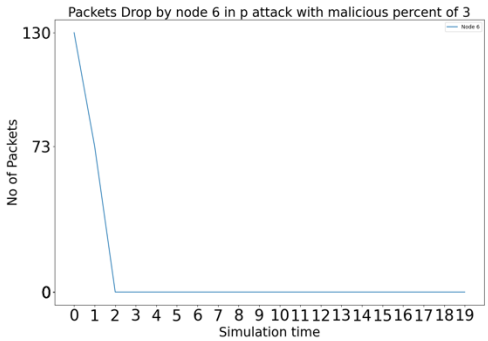

Fig 4.1.3.P.D.6: Packets Dropped by Node 6 with 3 attackers in Periodic attack.

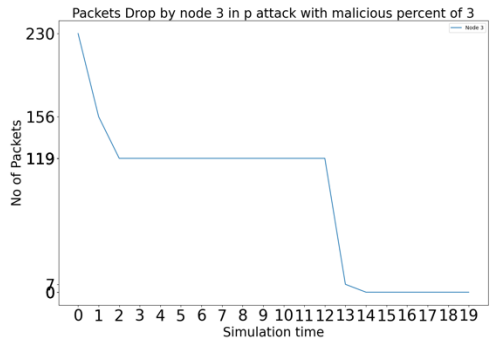

Fig 4.1.3.P.D.3: Packets Dropped by Node 3 with 3 attackers in Periodic attack.

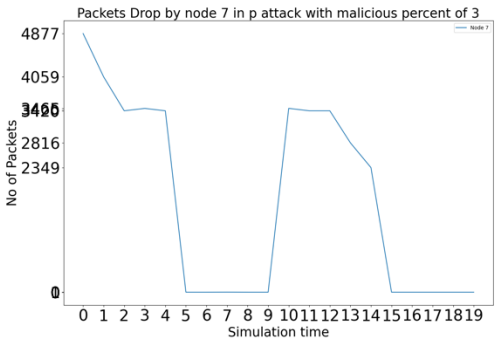

Fig 4.1.3.P.D.7: Packets Dropped by Node 7 with 3 attackers in Periodic attack.

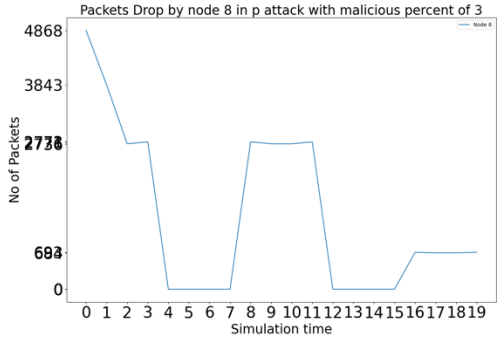

Fig 4.1.3.P.D.8: Packets Dropped by Node 8 with 3 attackers in Periodic attack.

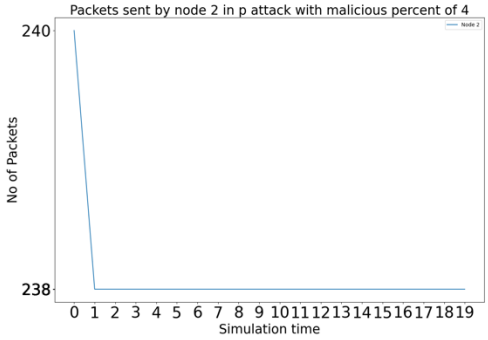

Fig 4.1.4.P.S.2: Packets Sent by Node 2 with 4 attackers in Periodic attack.

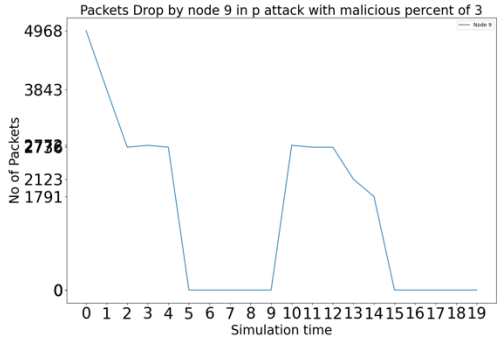

Fig 4.1.3.P.D.9: Packets Dropped by Node 9 with 3 attackers in Periodic attack.

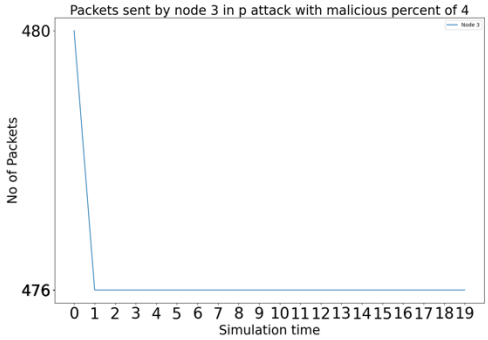

Fig 4.1.4.P.S.3: Packets Sent by Node 3 with 4 attackers in Periodic attack.

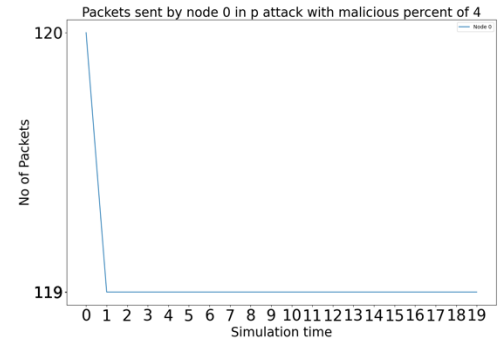

Fig 4.1.4.P.S.0: Packets Sent by Node 0 with 4 attackers in Periodic attack.

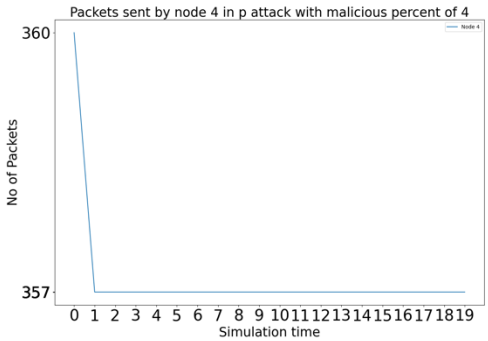

Fig 4.1.4.P.S.4: Packets Sent by Node 4 with 4 attackers in Periodic attack.

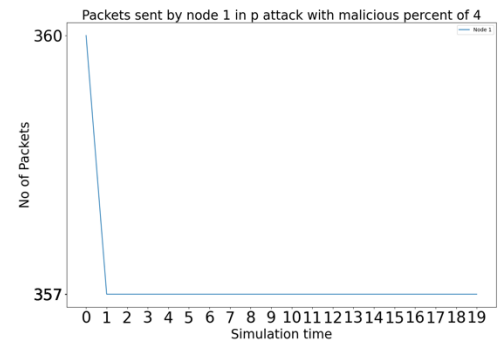

Fig 4.1.4.P.S.1: Packets Sent by Node 1 with 4 attackers in Periodic attack.

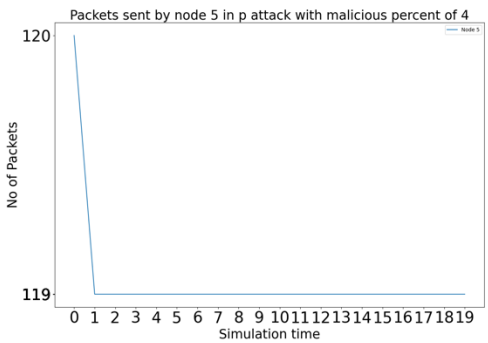

Fig 4.1.4.P.S.5: Packets Sent by Node 5 with 4 attackers in Periodic attack.

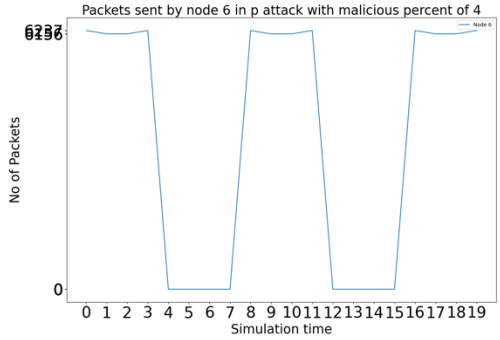

Fig 4.1.4.P.S.6: Packets Sent by Node 6 with 4 attackers in Periodic attack.

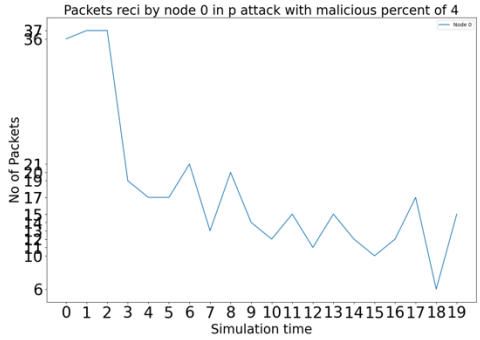

Fig 4.1.4.P.R.0: Packets Received by Node 0 with 4 attackers in Periodic attack.

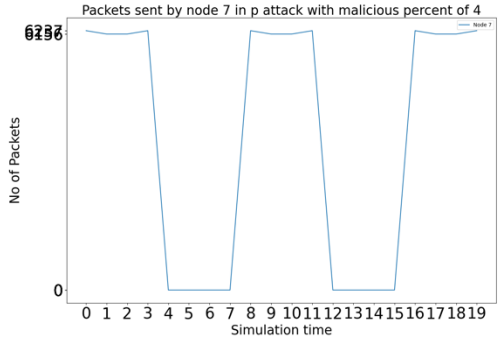

Fig 4.1.4.P.S.7: Packets Sent by Node 7 with 4 attackers in Periodic attack.

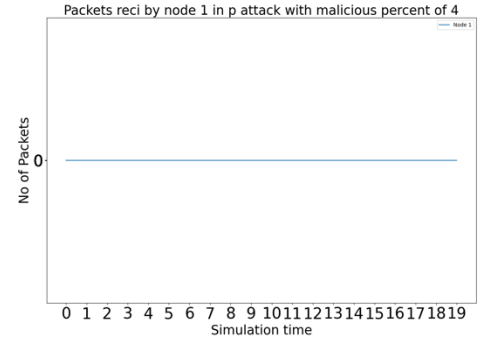

Fig 4.1.4.P.R.1: Packets Received by Node 1 with 4 attackers in Periodic attack.

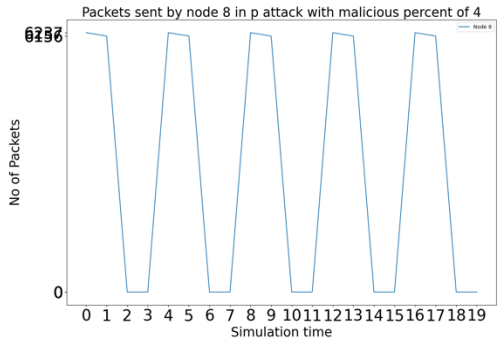

Fig 4.1.4.P.S.8: Packets Sent by Node 8 with 4 attackers in Periodic attack.

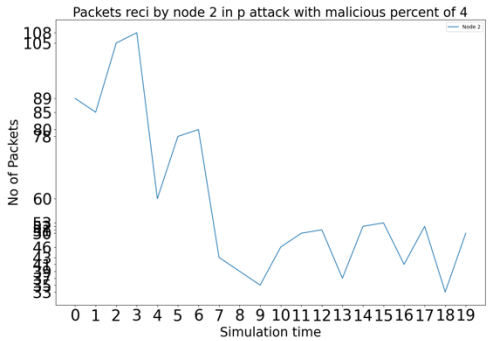

Fig 4.1.4.P.R.2: Packets Received by Node 2 with 4 attackers in Periodic attack.

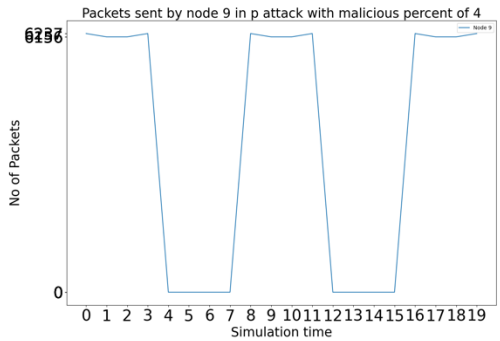

Fig 4.1.4.P.S.9: Packets Sent by Node 9 with 4 attackers in Periodic attack.

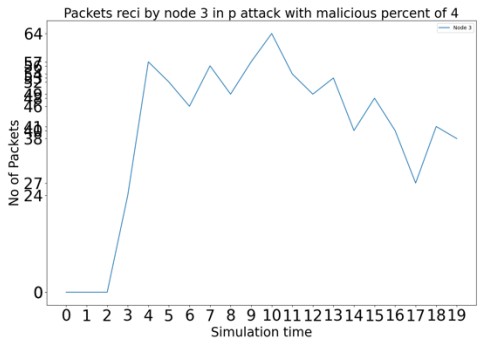

Fig 4.1.4.P.R.3: Packets Received by Node 3 with 4 attackers in Periodic attack.

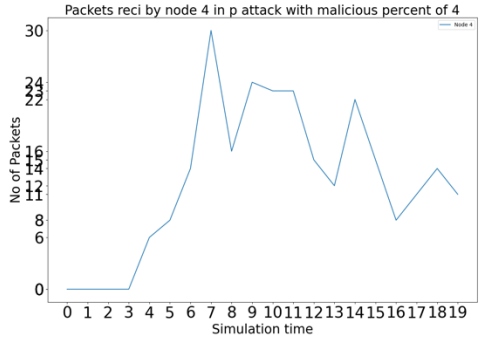

Fig 4.1.4.P.R.4: Packets Received by Node 4 with 4 attackers in Periodic attack.

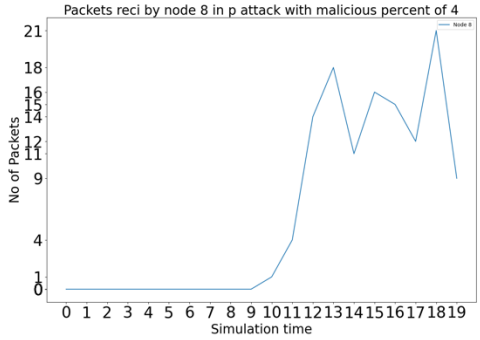

Fig 4.1.4.P.R.8: Packets Received by Node 8 with 4 attackers in Periodic attack.

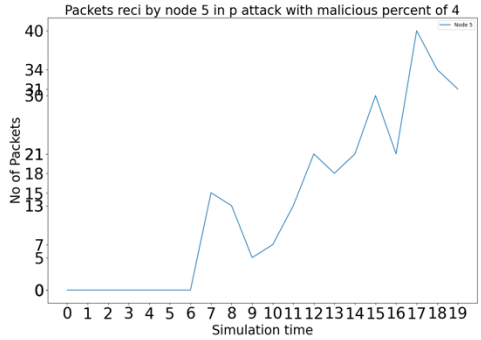

Fig 4.1.4.P.R.5: Packets Received by Node 5 with 4 attackers in Periodic attack.

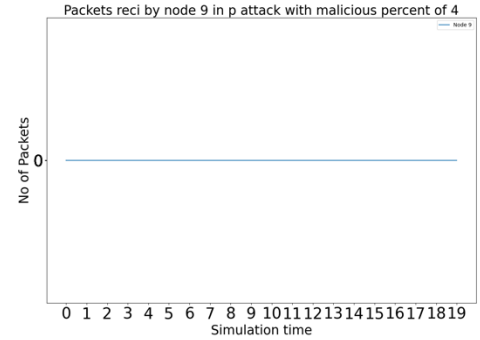

Fig 4.1.4.P.R.9: Packets Received by Node 9 with 4 attackers in Periodic attack.

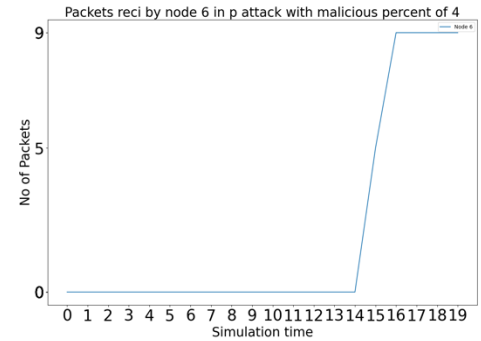

Fig 4.1.4.P.R.6: Packets Received by Node 6 with 4 attackers in Periodic attack.

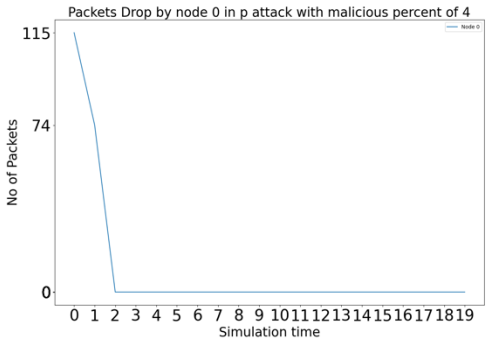

Fig 4.1.4.P.D.0: Packets Dropped by Node 0 with 4 attackers in Periodic attack.

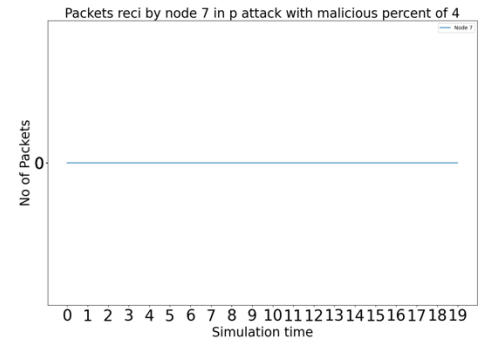

Fig 4.1.4.P.R.7: Packets Received by Node 7 with 4 attackers in Periodic attack.

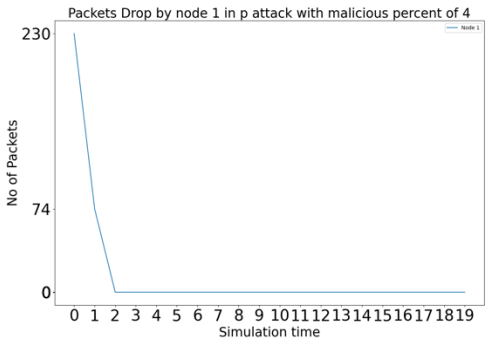

Fig 4.1.4.P.D.1: Packets Dropped by Node 1 with 4 attackers in Periodic attack.

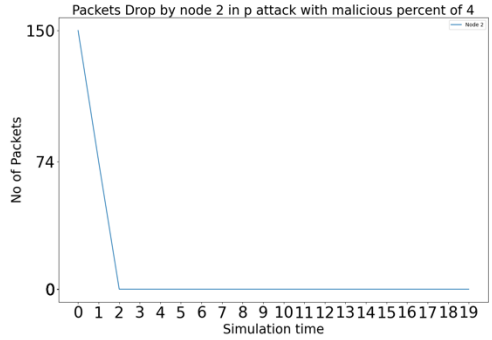

Fig 4.1.4.P.D.2: Packets Dropped by Node 2 with 4 attackers in Periodic attack.

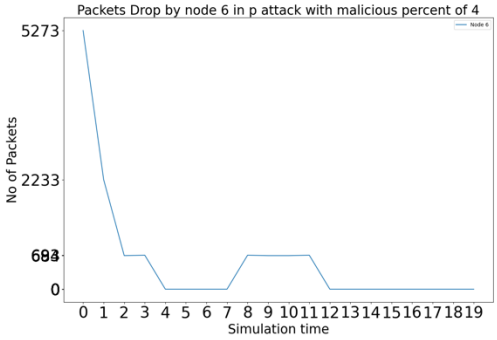

Fig 4.1.4.P.D.6: Packets Dropped by Node 6 with 4 attackers in Periodic attack.

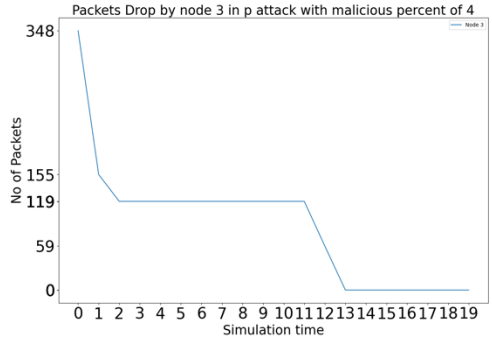

Fig 4.1.4.P.D.3: Packets Dropped by Node 3 with 4 attackers in Periodic attack.

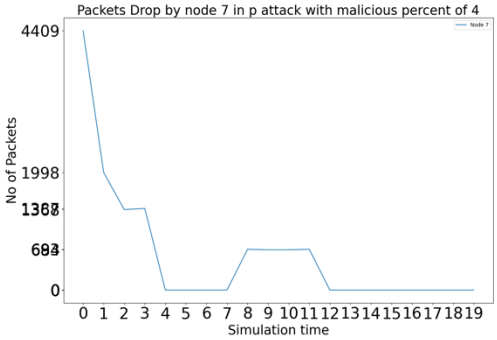

Fig 4.1.4.P.D.7: Packets Dropped by Node 7 with 4 attackers in Periodic attack.

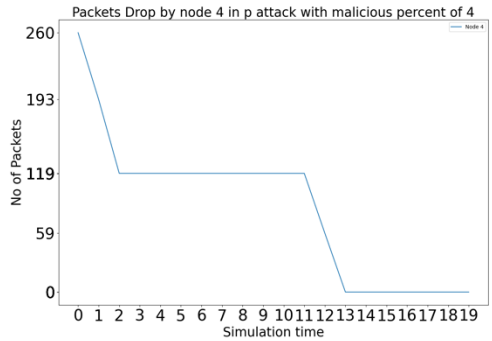

Fig 4.1.4.P.D.4: Packets Dropped by Node 4 with 4 attackers in Periodic attack.

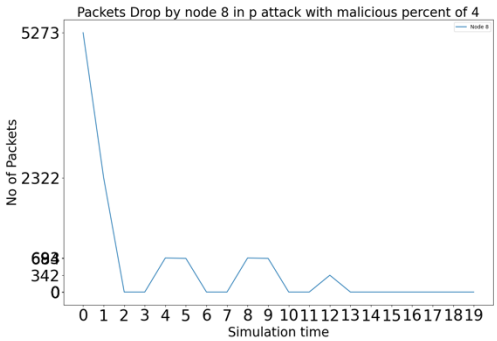

Fig 4.1.4.P.D.8: Packets Dropped by Node 8 with 4 attackers in Periodic attack.

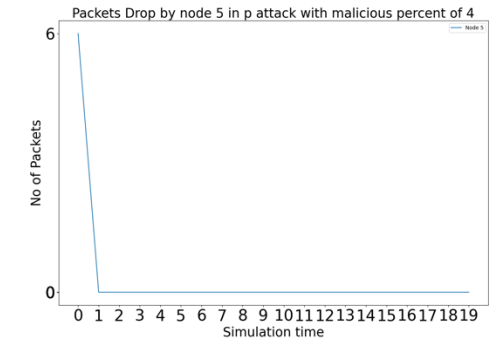

Fig 4.1.4.P.D.5: Packets Dropped by Node 5 with 4 attackers in Periodic attack.

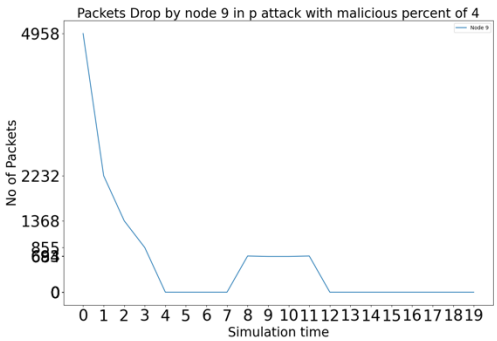

Fig 4.1.4.P.D.9: Packets Dropped by Node 9 with 4 attackers in Periodic attack.

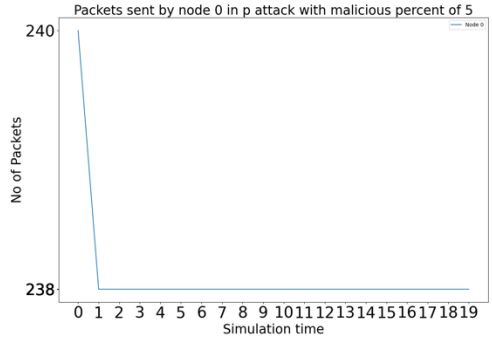

Fig 4.1.5.P.S.0: Packets Sent by Node 0 with 5 attackers in Periodic attack.

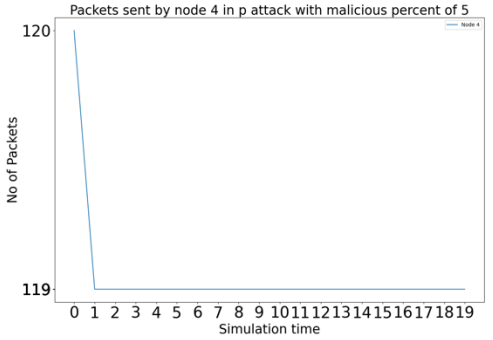

Fig 4.1.5.P.S.4: Packets Sent by Node 4 with 5 attackers in Periodic attack.

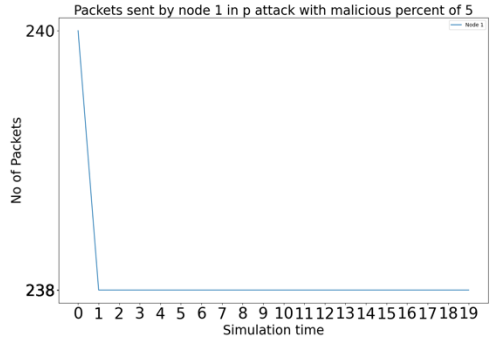

Fig 4.1.5.P.S.1: Packets Sent by Node 1 with 5 attackers in Periodic attack.

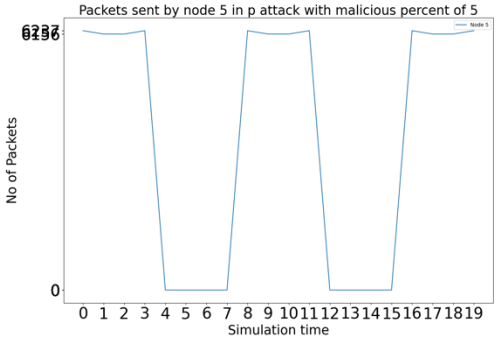

Fig 4.1.5.P.S.5: Packets Sent by Node 5 with 5 attackers in Periodic attack.

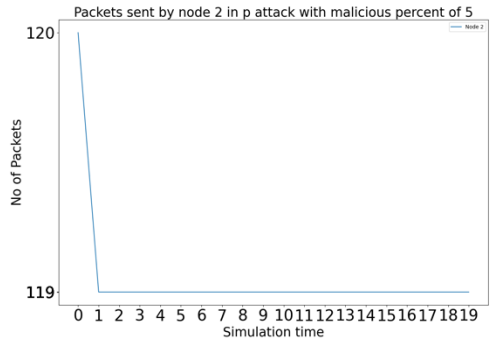

Fig 4.1.5.P.S.2: Packets Sent by Node 2 with 5 attackers in Periodic attack.

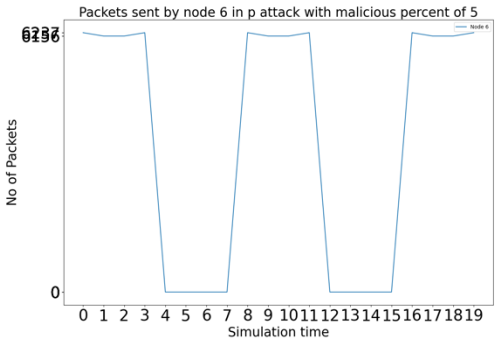

Fig 4.1.5.P.S.6: Packets Sent by Node 6 with 5 attackers in Periodic attack.

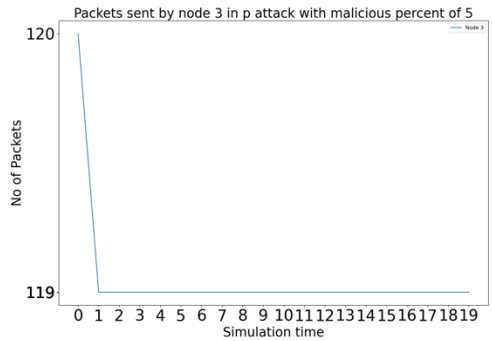

Fig 4.1.5.P.S.3: Packets Sent by Node 3 with 5 attackers in Periodic attack.

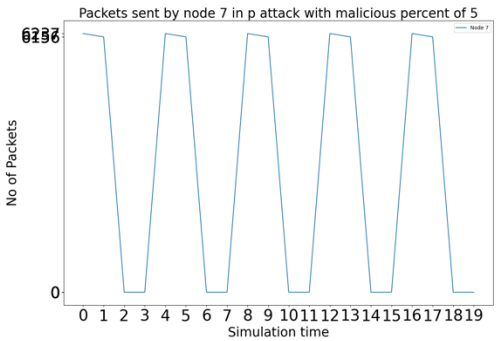

Fig 4.1.5.P.S.7: Packets Sent by Node 7 with 5 attackers in Periodic attack.

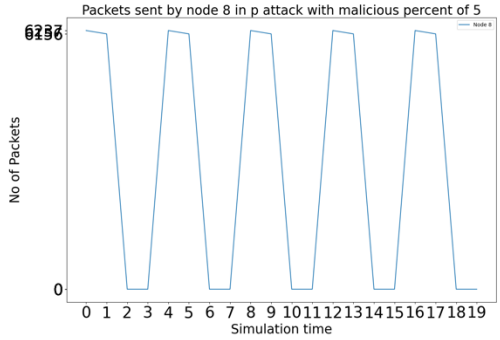

Fig 4.1.5.P.S.8: Packets Sent by Node 8 with 5 attackers in Periodic attack.

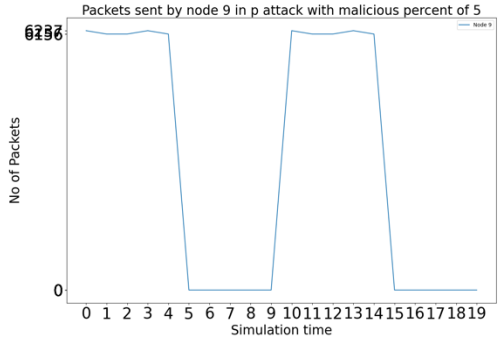

Fig 4.1.5.P.S.9: Packets Sent by Node 9 with 5 attackers in Periodic attack.

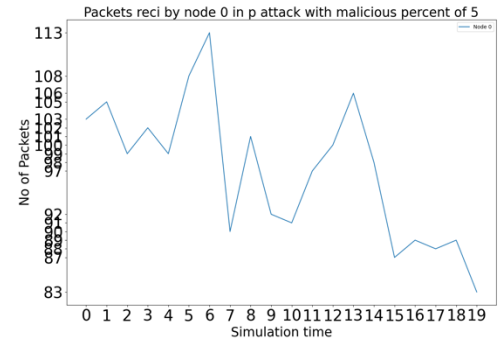

Fig 4.1.5.P.R.0: Packets Received by Node 0 with 5 attackers in Periodic attack.

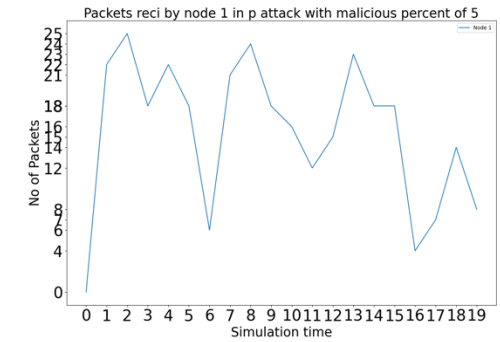

Fig 4.1.5.P.R.1: Packets Received by Node 1 with 5 attackers in Periodic attack.

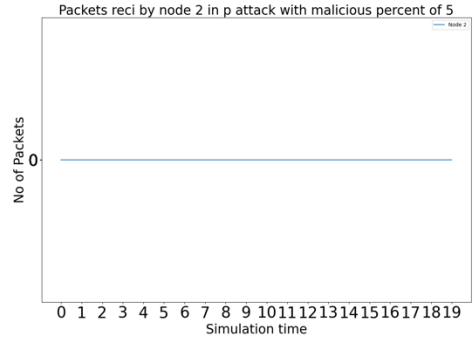

Fig 4.1.5.P.R.2: Packets Received by Node 2 with 5 attackers in Periodic attack.

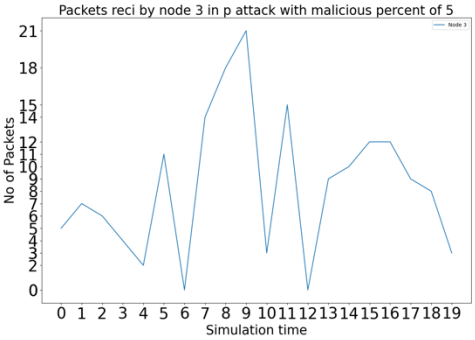

Fig 4.1.5.P.R.3: Packets Received by Node 3 with 5 attackers in Periodic attack.

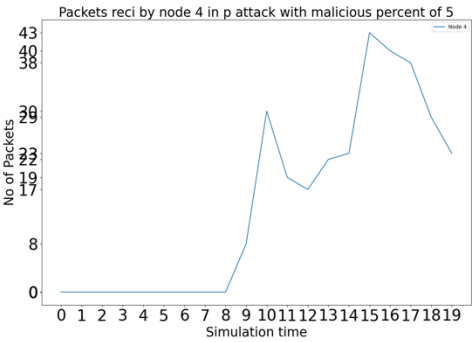

Fig 4.1.5.P.R.4: Packets Received by Node 4 with 5 attackers in Periodic attack.

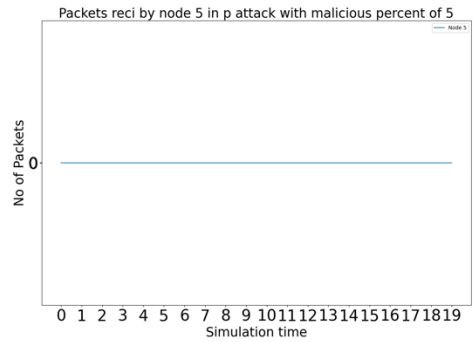

Fig 4.1.5.P.R.5: Packets Received by Node 5 with 5 attackers in Periodic attack.

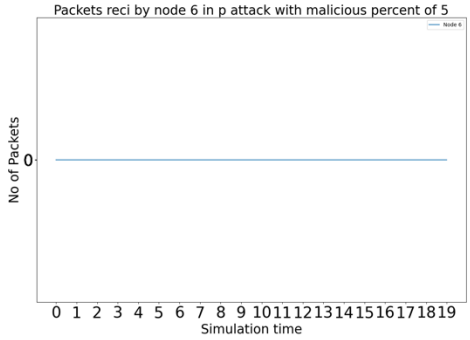

Fig 4.1.5.P.R.6: Packets Received by Node 6 with 5 attackers in Periodic attack.

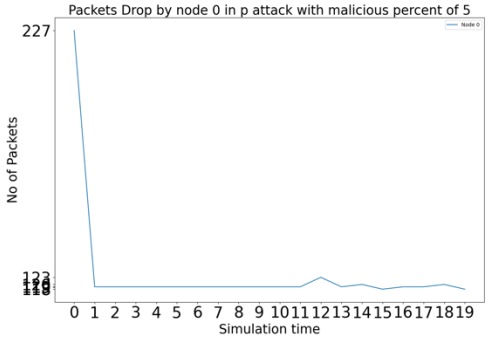

Fig 4.1.5.P.D.0: Packets Dropped by Node 0 with 5 attackers in Periodic attack.

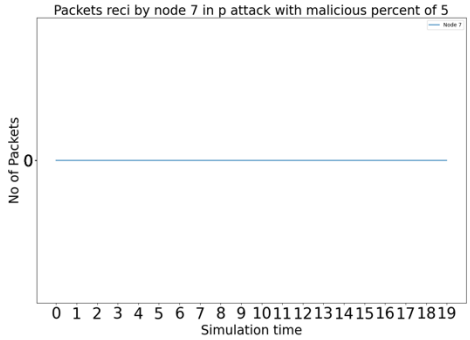

Fig 4.1.5.P.R.7: Packets Received by Node 7 with 5 attackers in Periodic attack.

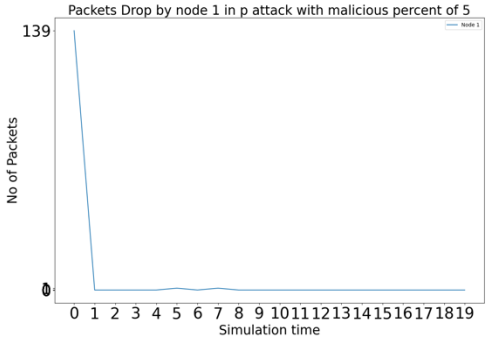

Fig 4.1.5.P.D.1: Packets Dropped by Node 1 with 5 attackers in Periodic attack.

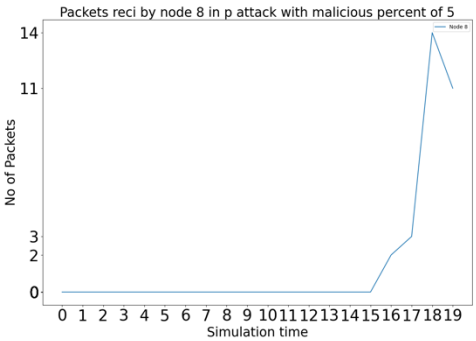

Fig 4.1.5.P.R.8: Packets Received by Node 8 with 5 attackers in Periodic attack.

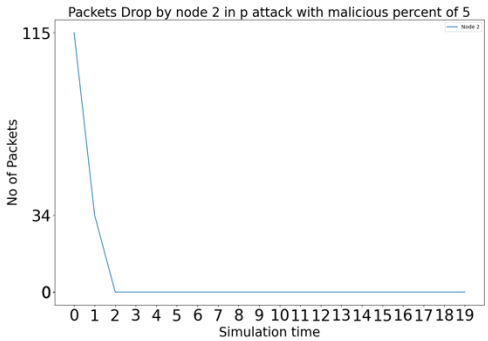

Fig 4.1.5.P.D.2: Packets Dropped by Node 2 with 5 attackers in Periodic attack.

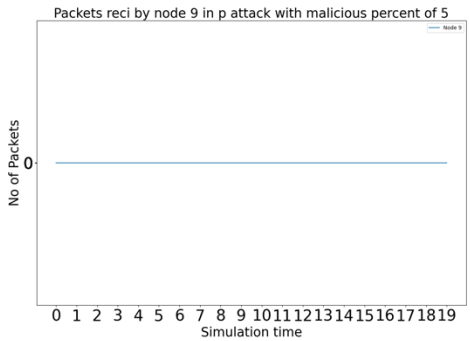

Fig 4.1.5.P.R.9: Packets Received by Node 9 with 5 attackers in Periodic attack.

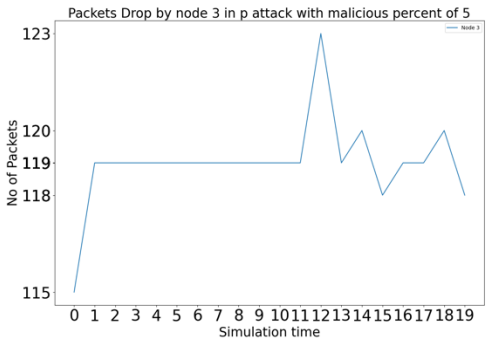

Fig 4.1.5.P.D.3: Packets Dropped by Node 3 with 5 attackers in Periodic attack.

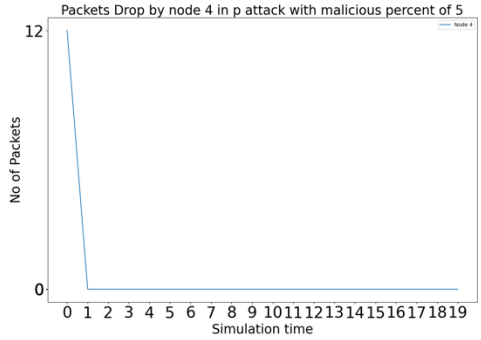

Fig 4.1.5.P.D.4: Packets Dropped by Node 4 with 5 attackers in Periodic attack.

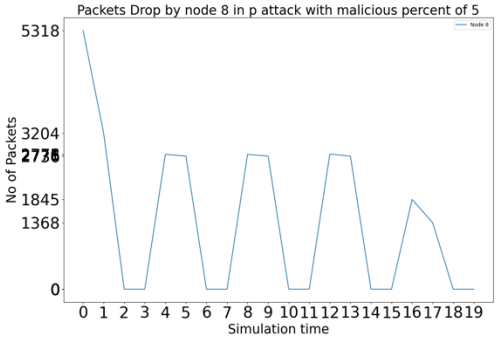

Fig 4.1.5.P.D.8: Packets Dropped by Node 8 with 5 attackers in Periodic attack.

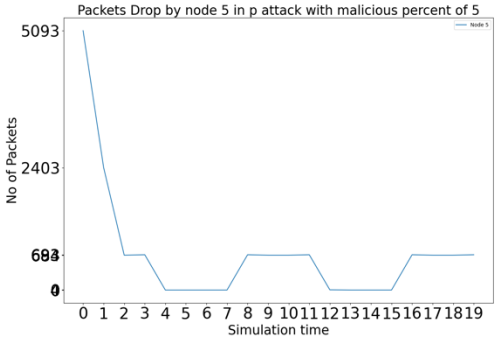

Fig 4.1.5.P.D.5: Packets Dropped by Node 5 with 5 attackers in Periodic attack.

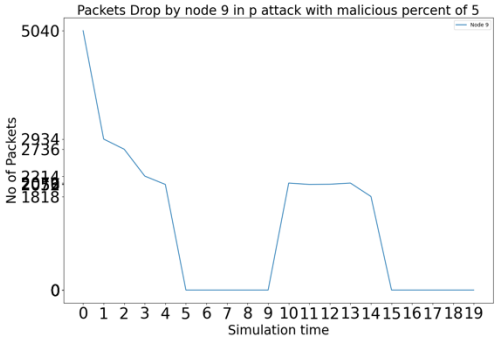

Fig 4.1.5.P.D.9: Packets Dropped by Node 9 with 5 attackers in Periodic attack.

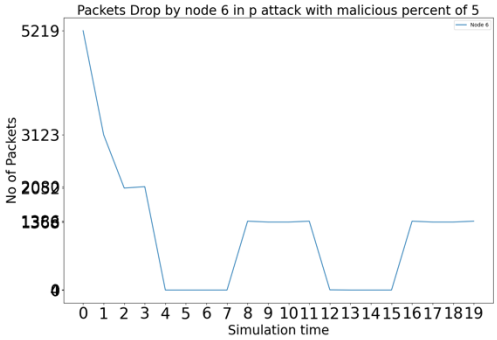

Fig 4.1.5.P.D.6: Packets Dropped by Node 6 with 5 attackers in Periodic attack.

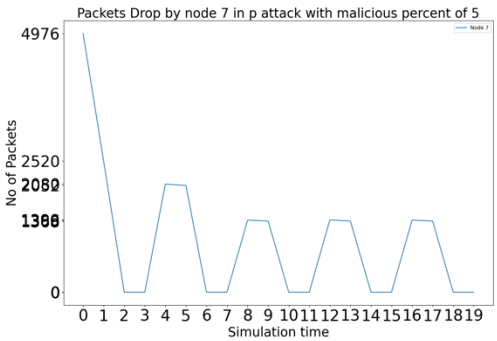

Fig 4.1.5.P.D.7: Packets Dropped by Node 7 with 5 attackers in Periodic attack.

Random Attack:

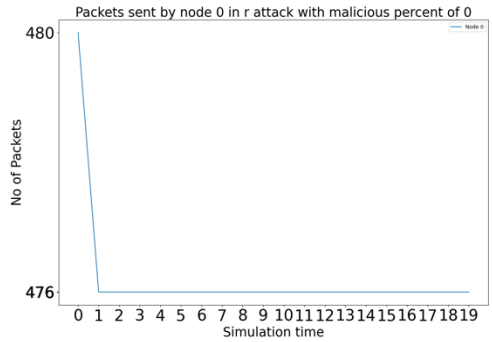

Fig 4.1.0.R.S.0: Packets Sent by Node 0 with 0 attackers in Random attack.

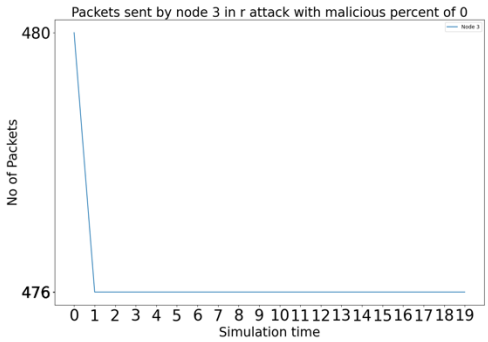

Fig 4.1.0.R.S.3: Packets Sent by Node 3 with 0 attackers in Random attack.

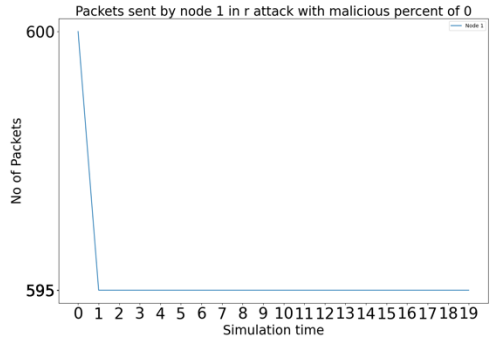

Fig 4.1.0.R.S.1: Packets Sent by Node 1 with 0 attackers in Random attack.

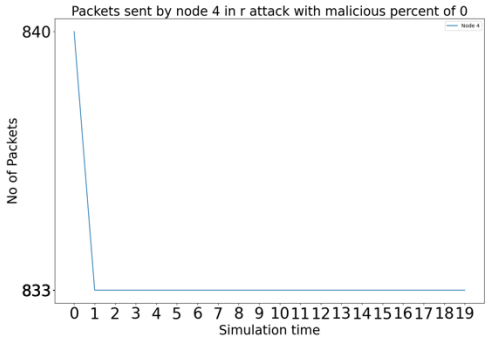

Fig 4.1.0.R.S.4: Packets Sent by Node 4 with 0 attackers in Random attack.

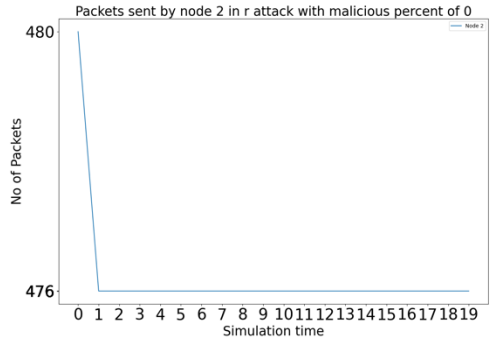

Fig 4.1.0.R.S.2: Packets Sent by Node 2 with 0 attackers in Random attack.

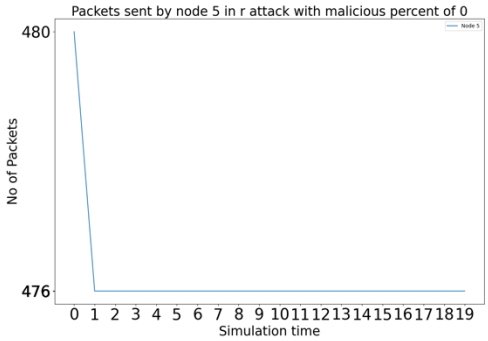

Fig 4.1.0.R.S.5: Packets Sent by Node 5 with 0 attackers in Random attack.

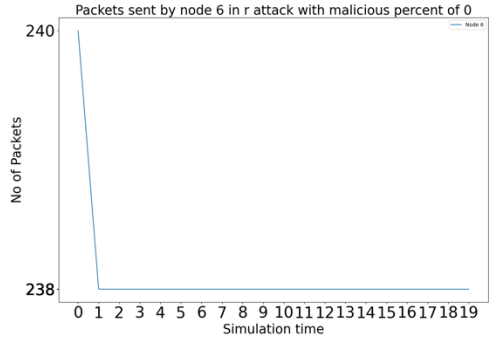

Fig 4.1.0.R.S.6: Packets Sent by Node 6 with 0 attackers in Random attack.

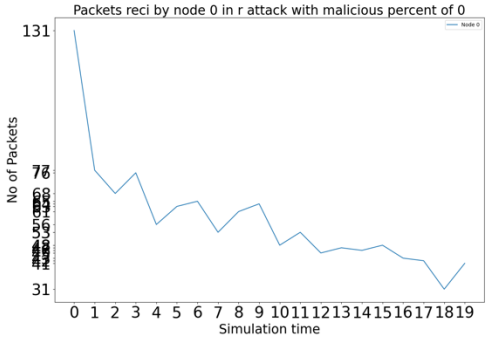

Fig 4.1.0.R.R.0: Packets Received by Node 0 with 0 attackers in Random attack.

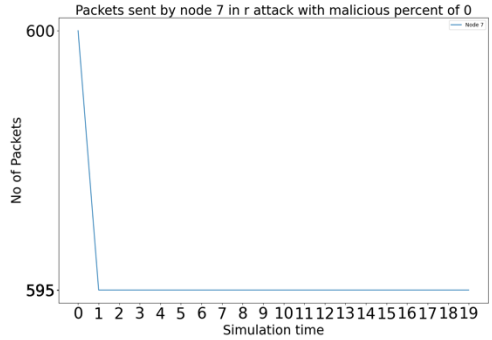

Fig 4.1.0.R.S.7: Packets Sent by Node 7 with 0 attackers in Random attack.

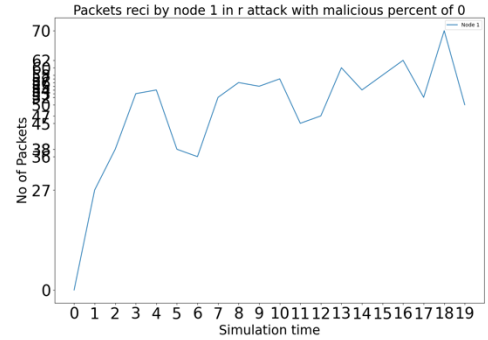

Fig 4.1.0.R.R.1: Packets Received by Node 1 with 0 attackers in Random attack.

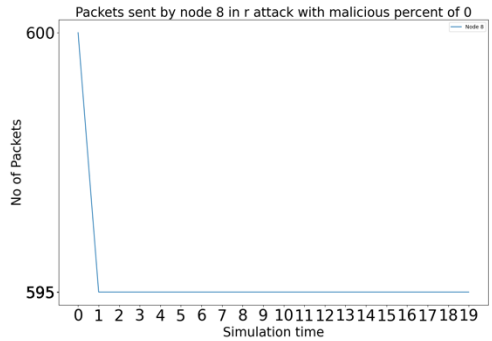

Fig 4.1.0.R.S.8: Packets Sent by Node 8 with 0 attackers in Random attack.

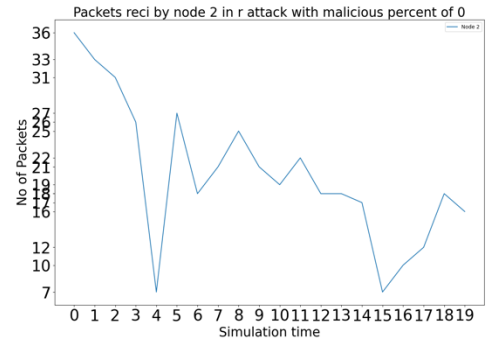

Fig 4.1.0.R.R.2: Packets Received by Node 2 with 0 attackers in Random attack.

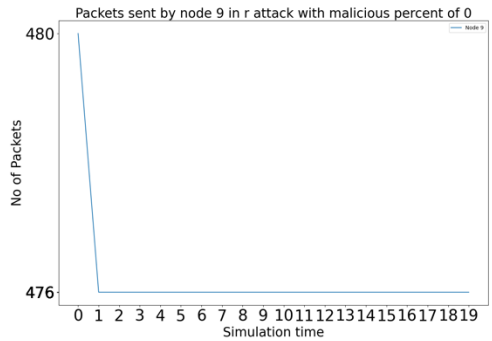

Fig 4.1.0.R.S.9: Packets Sent by Node 9 with 0 attackers in Random attack.

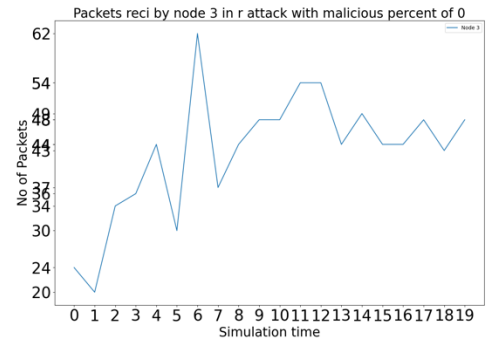

Fig 4.1.0.R.R.3: Packets Received by Node 3 with 0 attackers in Random attack.

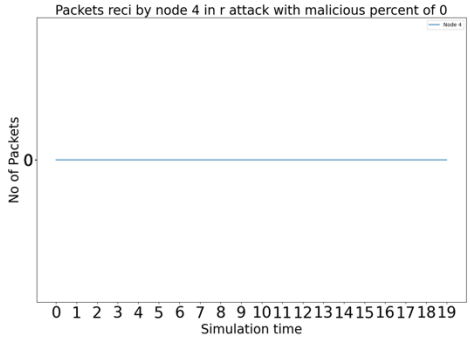

Fig 4.1.0.R.R.4: Packets Received by Node 4 with 0 attackers in Random attack.

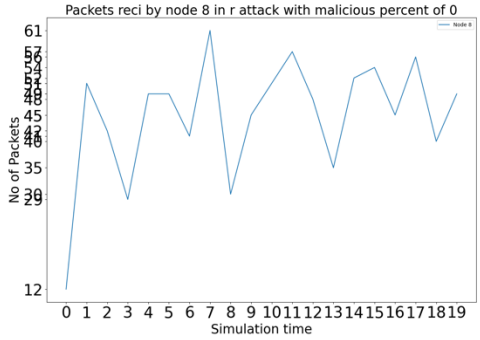

Fig 4.1.0.R.R.8: Packets Received by Node 8 with 0 attackers in Random attack.

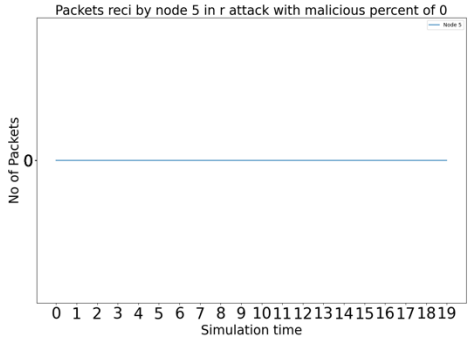

Fig 4.1.0.R.R.5: Packets Received by Node 5 with 0 attackers in Random attack.

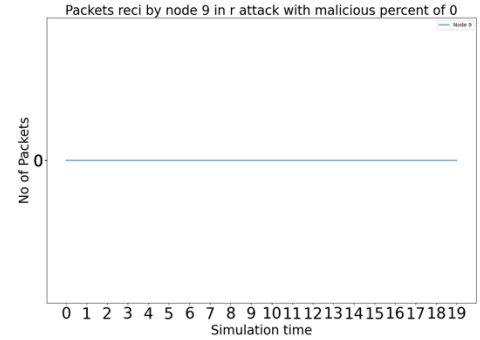

Fig 4.1.0.R.R.9: Packets Received by Node 9 with 0 attackers in Random attack.

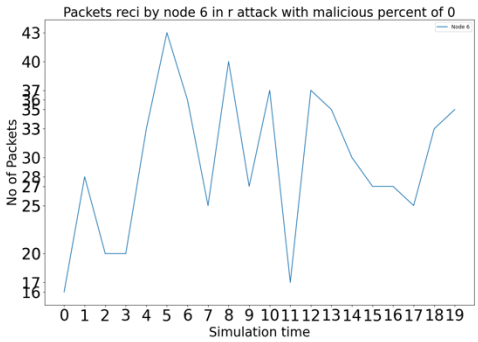

Fig 4.1.0.R.R.6: Packets Received by Node 6 with 0 attackers in Random attack.

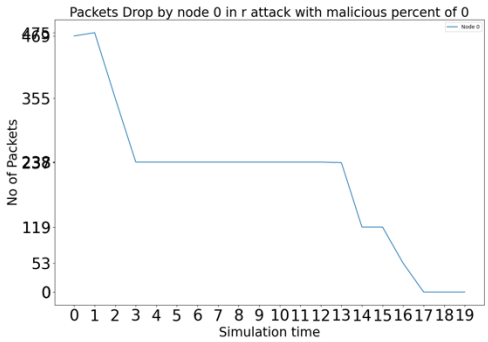

Fig 4.1.0.R.D.0: Packets Dropped by Node 0 with 0 attackers in Random attack.

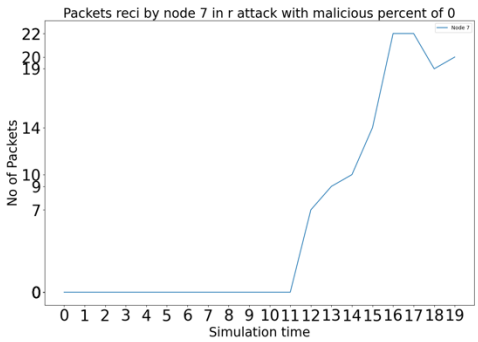

Fig 4.1.0.R.R.7: Packets Received by Node 7 with 0 attackers in Random attack.

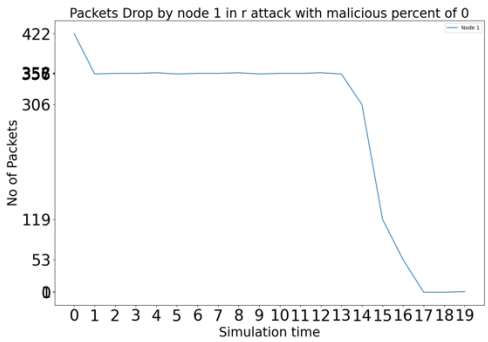

Fig 4.1.0.R.D.1: Packets Dropped by Node 1 with 0 attackers in Random attack.

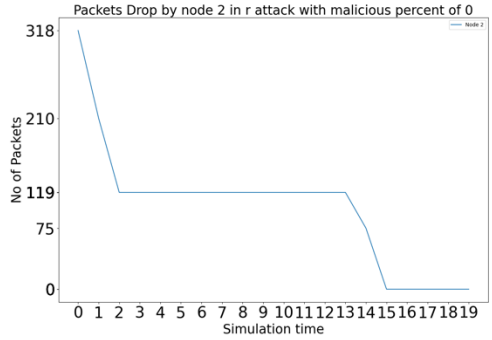

Fig 4.1.0.R.D.2: Packets Dropped by Node 2 with 0 attackers in Random attack.

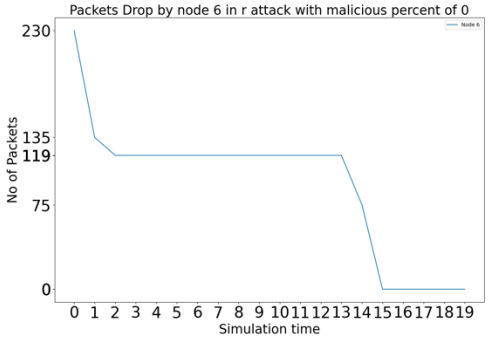

Fig 4.1.0.R.D.6: Packets Dropped by Node 6 with 0 attackers in Random attack.

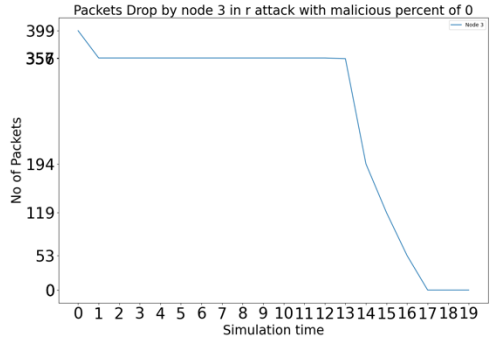

Fig 4.1.0.R.D.3: Packets Dropped by Node 3 with 0 attackers in Random attack.

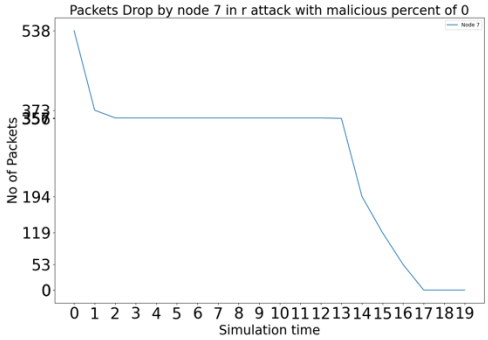

Fig 4.1.0.R.D.7: Packets Dropped by Node 7 with 0 attackers in Random attack.

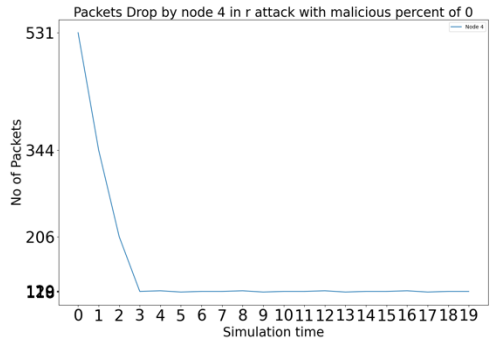

Fig 4.1.0.R.D.4: Packets Dropped by Node 4 with 0 attackers in Random attack.

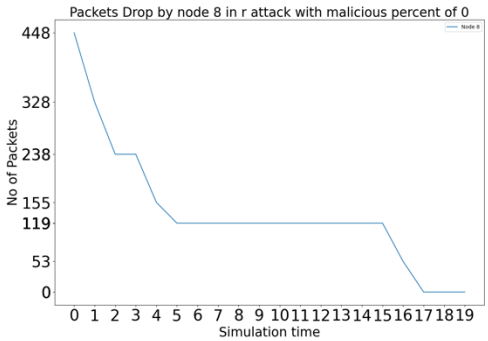

Fig 4.1.0.R.D.8: Packets Dropped by Node 8 with 0 attackers in Random attack.

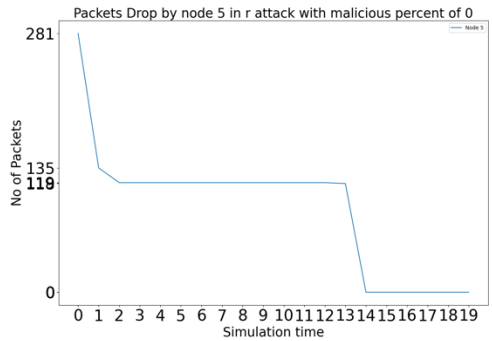

Fig 4.1.0.R.D.5: Packets Dropped by Node 5 with 0 attackers in Random attack.

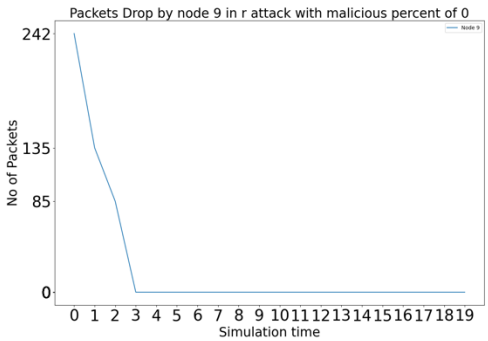

Fig 4.1.0.R.D.9: Packets Dropped by Node 9 with 0 attackers in Random attack.

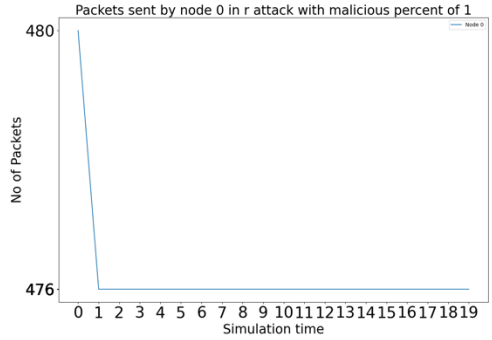

Fig 4.1.1.R.S.0: Packets Sent by Node 0 with 1 attacker in Random attack.

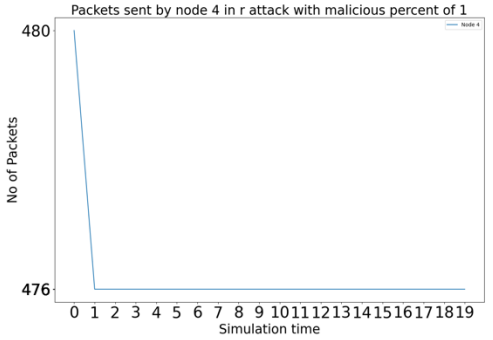

Fig 4.1.1.R.S.4: Packets Sent by Node 4 with 1 attacker in Random attack.

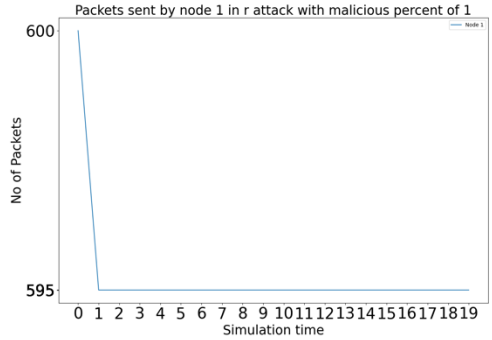

Fig 4.1.1.R.S.1: Packets Sent by Node 1 with 1 attacker in Random attack.

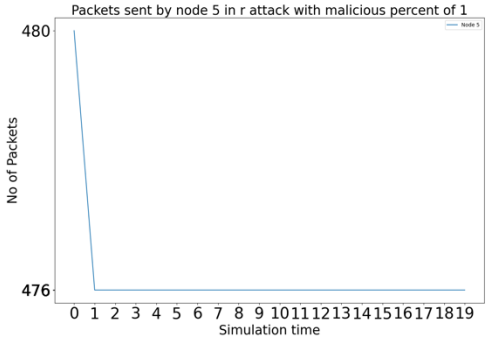

Fig 4.1.1.R.S.5: Packets Sent by Node 5 with 1 attacker in Random attack.

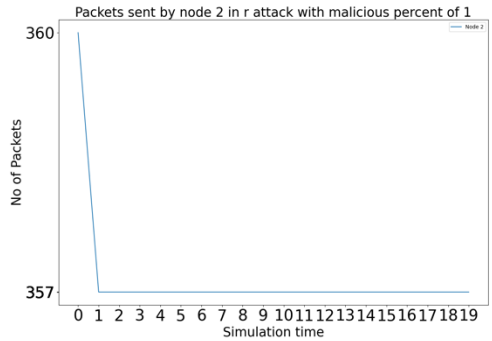

Fig 4.1.1.R.S.2: Packets Sent by Node 2 with 1 attacker in Random attack.

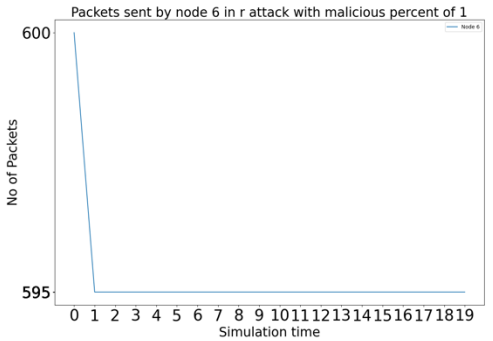

Fig 4.1.1.R.S.6: Packets Sent by Node 6 with 1 attacker in Random attack.

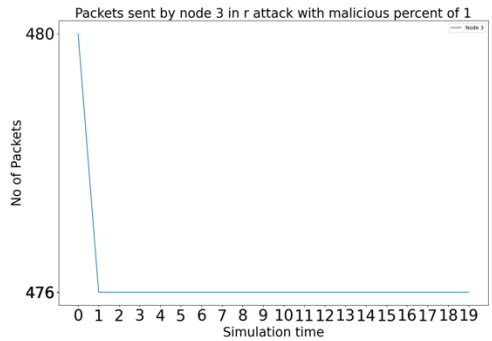

Fig 4.1.1.R.S.3: Packets Sent by Node 3 with 1 attacker in Random attack.

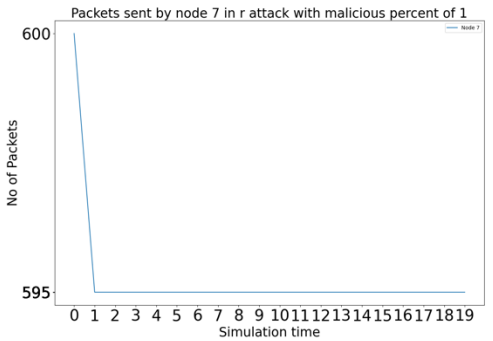

Fig 4.1.1.R.S.7: Packets Sent by Node 7 with 1 attacker in Random attack.

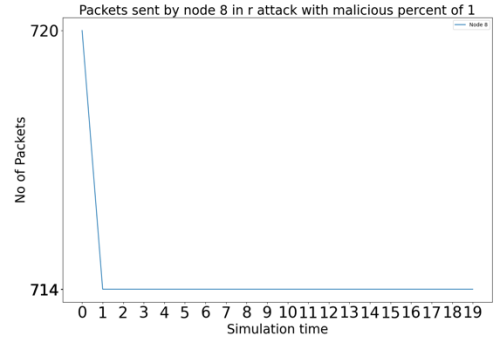

Fig 4.1.1.R.S.8: Packets Sent by Node 8 with 1 attacker in Random attack.

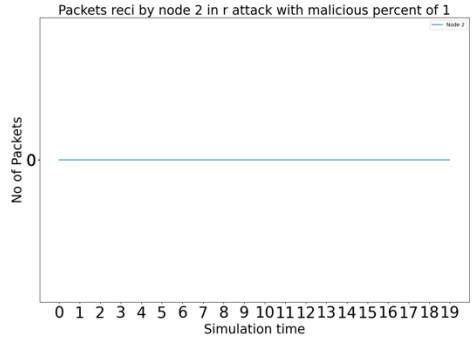

Fig 4.1.1.R.R.2: Packets Received by Node 2 with 1 attacker in Random attack.

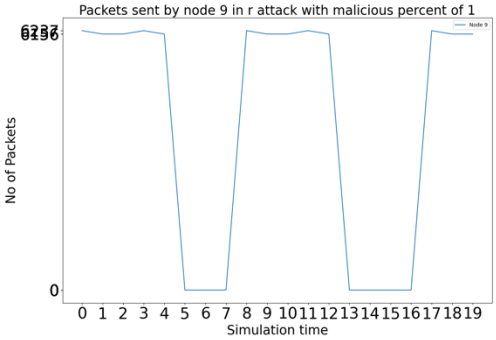

Fig 4.1.1.R.S.9: Packets Sent by Node 9 with 1 attacker in Random attack.

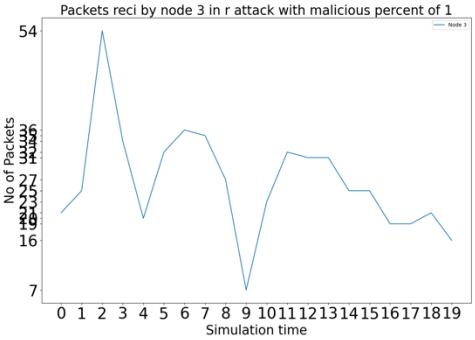

Fig 4.1.1.R.R.3: Packets Received by Node 3 with 1 attacker in Random attack.

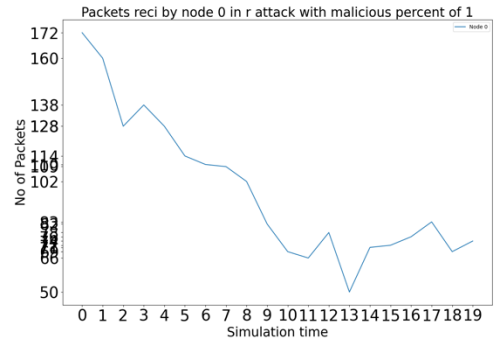

Fig 4.1.1.R.R.0: Packets Received by Node 0 with 1 attacker in Random attack.

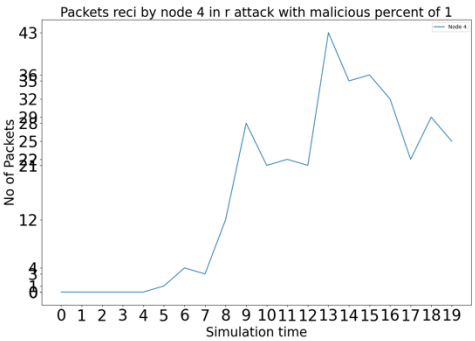

Fig 4.1.1.R.R.4: Packets Received by Node 4 with 1 attacker in Random attack.

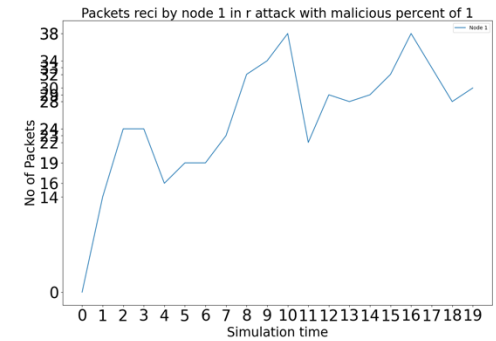

Fig 4.1.1.R.R.1: Packets Received by Node 1 with 1 attacker in Random attack.

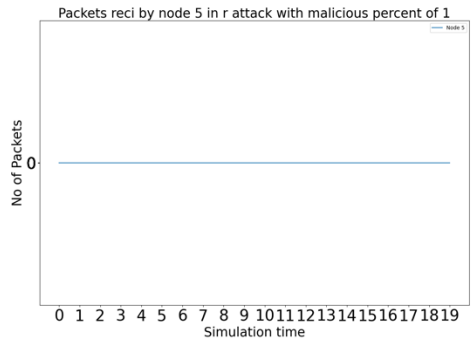

Fig 4.1.1.R.R.5: Packets Received by Node 5 with 1 attacker in Random attack.

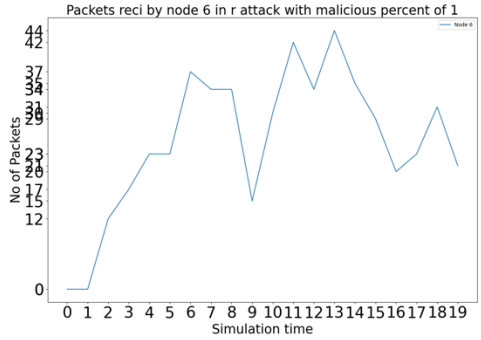

Fig 4.1.1.R.R.6: Packets Received by Node 6 with 1 attacker in Random attack.

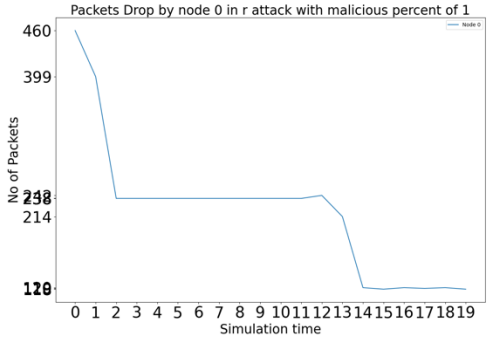

Fig 4.1.1.R.D.0: Packets Dropped by Node 0 with 1 attacker in Random attack.

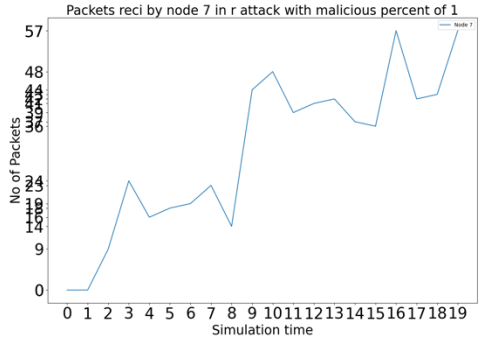

Fig 4.1.1.R.R.7: Packets Received by Node 7 with 1 attacker in Random attack.

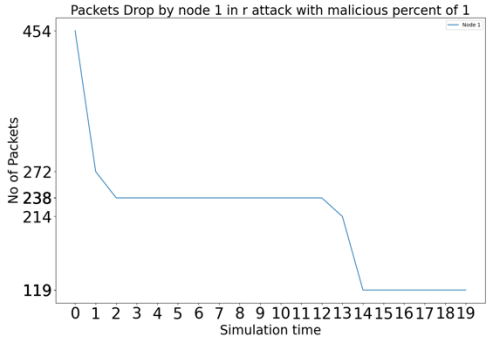

Fig 4.1.1.R.D.1: Packets Dropped by Node 1 with 1 attacker in Random attack.

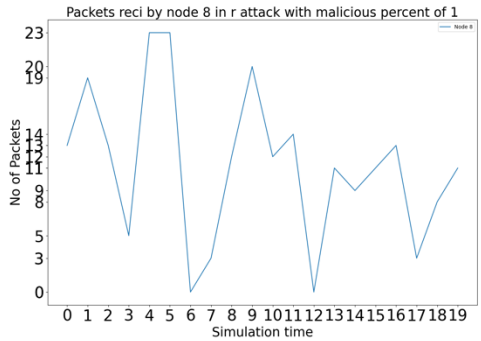

Fig 4.1.1.R.R.8: Packets Received by Node 8 with 1 attacker in Random attack.

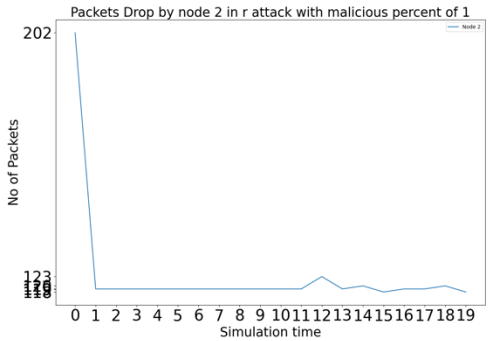

Fig 4.1.1.R.D.2: Packets Dropped by Node 2 with 1 attacker in Random attack.

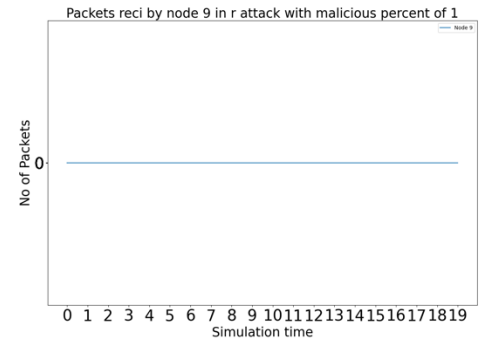

Fig 4.1.1.R.R.9: Packets Received by Node 9 with 1 attacker in Random attack.

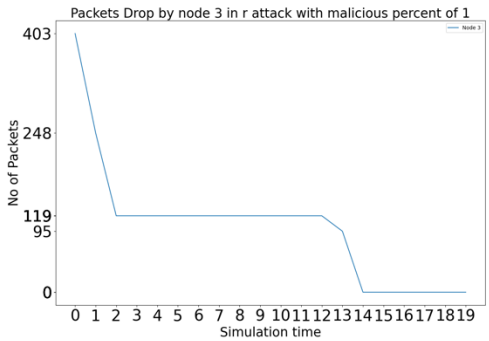

Fig 4.1.1.R.D.3: Packets Dropped by Node 3 with 1 attacker in Random attack.

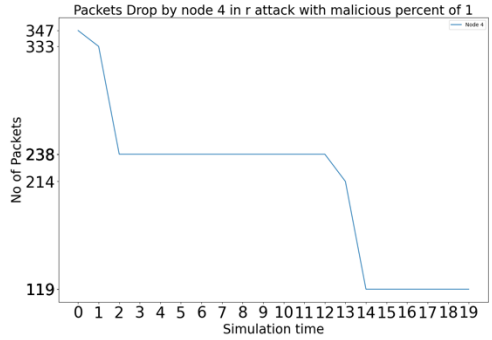

Fig 4.1.1.R.D.4: Packets Dropped by Node 4 with 1 attacker in Random attack.

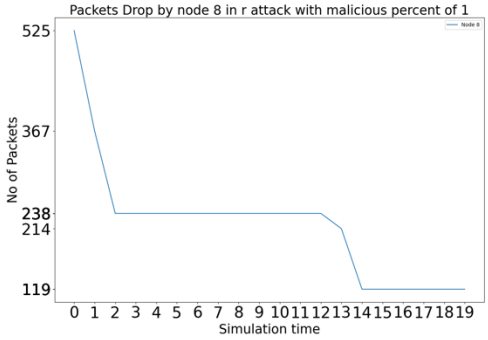

Fig 4.1.1.R.D.8: Packets Dropped by Node 8 with 1 attacker in Random attack.

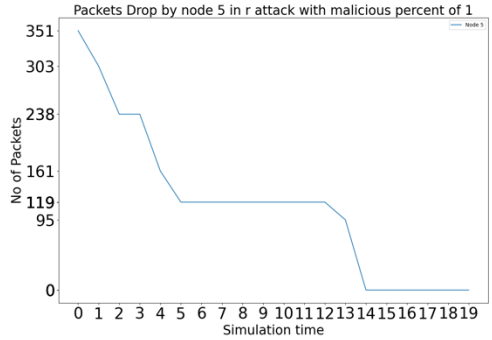

Fig 4.1.1.R.D.5: Packets Dropped by Node 5 with 1 attacker in Random attack.

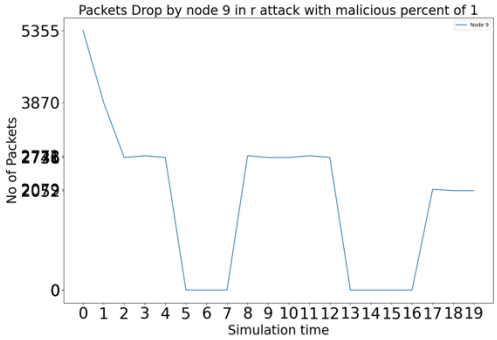

Fig 4.1.1.R.D.9: Packets Dropped by Node 9 with 1 attacker in Random attack.

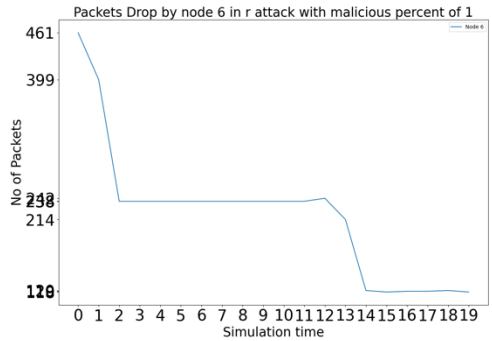

Fig 4.1.1.R.D.6: Packets Dropped by Node 6 with 1 attacker in Random attack.

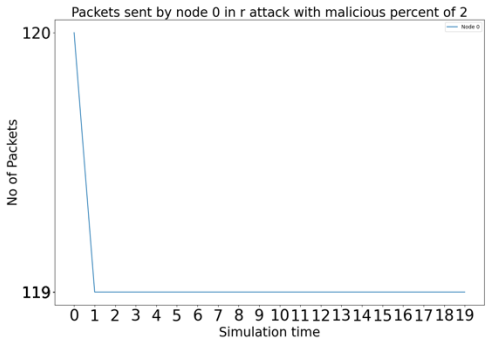

Fig 4.1.2.R.S.0: Packets Sent by Node 0 with 2 attackers in Random attack.

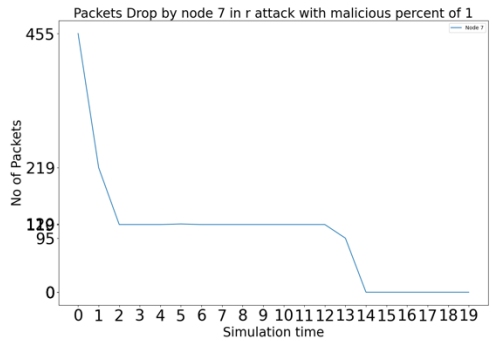

Fig 4.1.1.R.D.7: Packets Dropped by Node 7 with 1 attacker in Random attack.

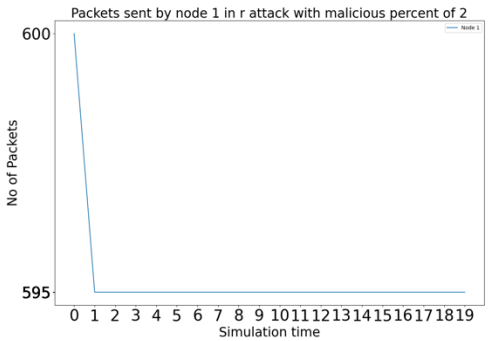

Fig 4.1.2.R.S.1: Packets Sent by Node 1 with 2 attackers in Random attack.

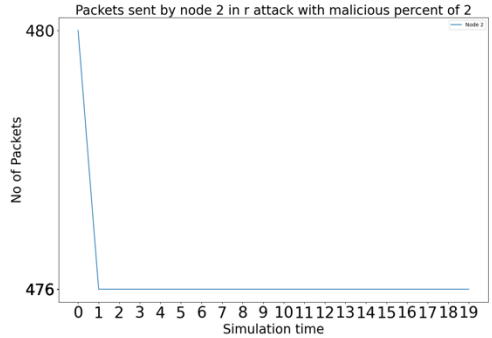

Fig 4.1.2.R.S.2: Packets Sent by Node 2 with 2 attackers in Random attack.

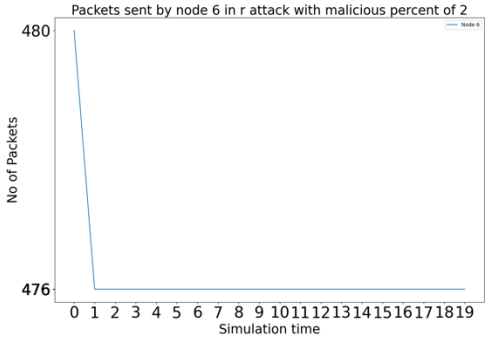

Fig 4.1.2.R.S.6: Packets Sent by Node 6 with 2 attackers in Random attack.

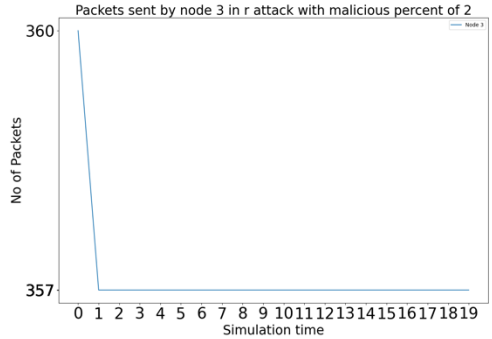

Fig 4.1.2.R.S.3: Packets Sent by Node 3 with 2 attackers in Random attack.

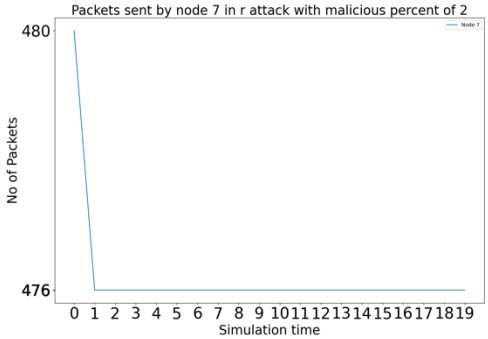

Fig 4.1.2.R.S.7: Packets Sent by Node 7 with 2 attackers in Random attack.

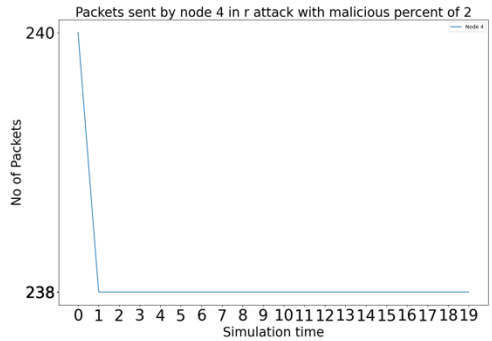

Fig 4.1.2.R.S.4: Packets Sent by Node 4 with 2 attackers in Random attack.

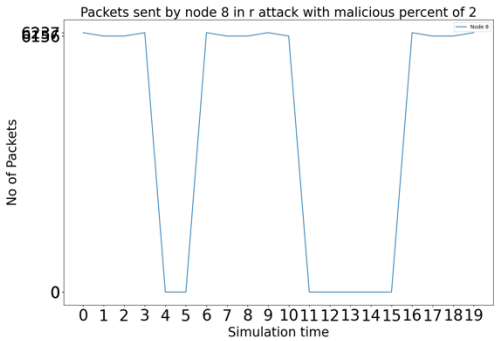

Fig 4.1.2.R.S.8: Packets Sent by Node 8 with 2 attackers in Random attack.

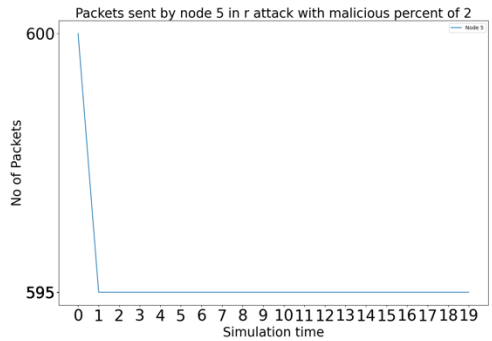

Fig 4.1.2.R.S.5: Packets Sent by Node 5 with 2 attackers in Random attack.

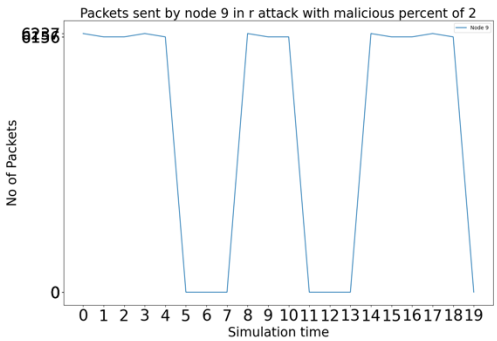

Fig 4.1.2.R.S.9: Packets Sent by Node 9 with 2 attackers in Random attack.

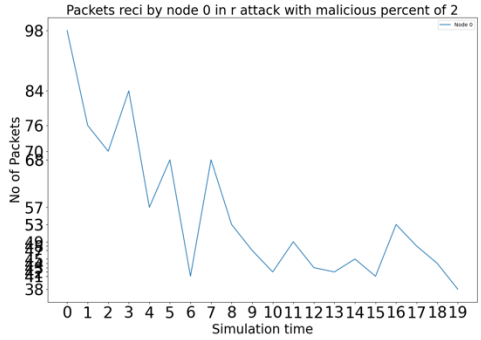

Fig 4.1.2.R.R.0: Packets Received by Node 0 with 2 attackers in Random attack.

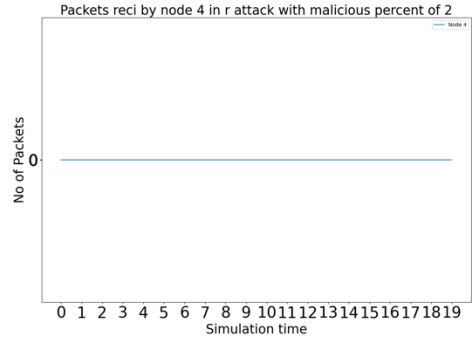

Fig 4.1.2.R.R.4: Packets Received by Node 4 with 2 attackers in Random attack.

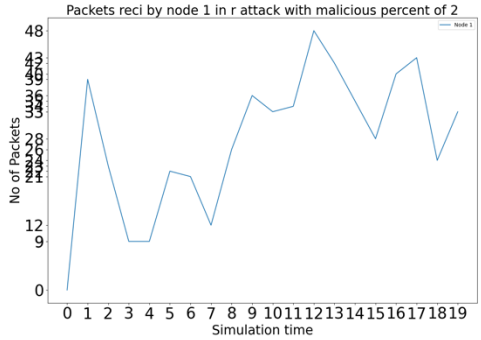

Fig 4.1.2.R.R.1: Packets Received by Node 1 with 2 attackers in Random attack.

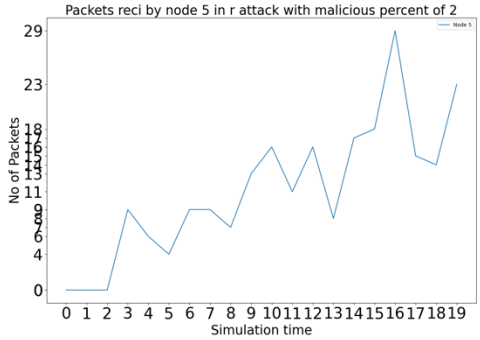

Fig 4.1.2.R.R.5: Packets Received by Node 5 with 2 attackers in Random attack.

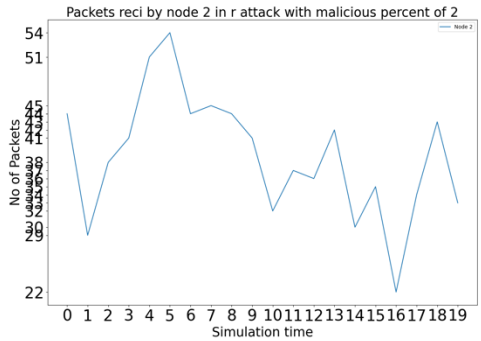

Fig 4.1.2.R.R.2: Packets Received by Node 2 with 2 attackers in Random attack.

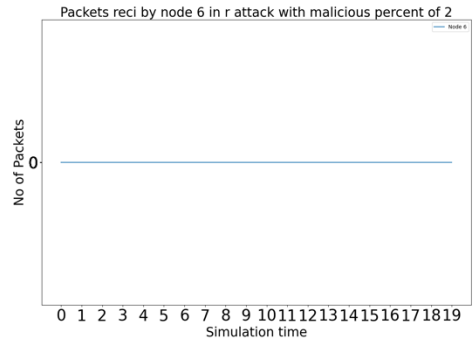

Fig 4.1.2.R.R.6: Packets Received by Node 6 with 2 attackers in Random attack.

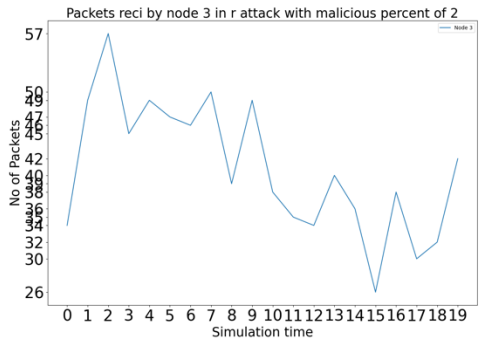

Fig 4.1.2.R.R.3: Packets Received by Node 3 with 2 attackers in Random attack.

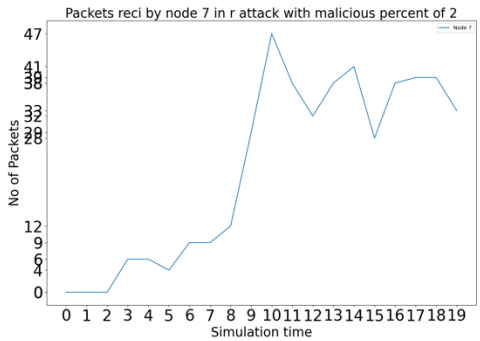

Fig 4.1.2.R.R.7: Packets Received by Node 7 with 2 attackers in Random attack.

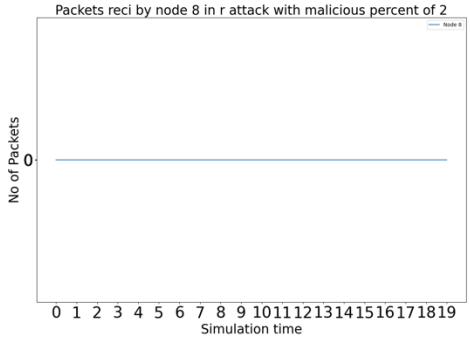

Fig 4.1.2.R.R.8: Packets Received by Node 8 with 2 attackers in Random attack.

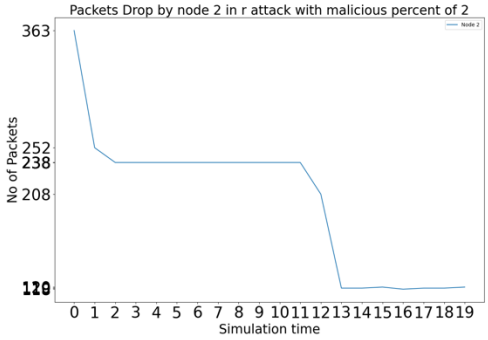

Fig 4.1.2.R.D.2: Packets Dropped by Node 2 with 2 attackers in Random attack.

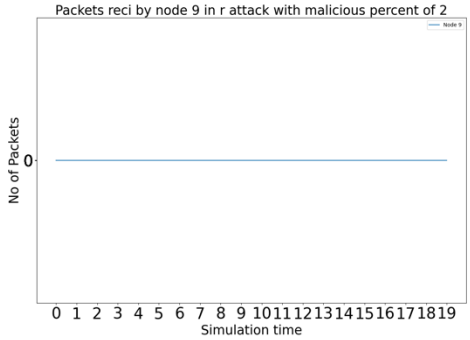

Fig 4.1.2.R.R.9: Packets Received by Node 9 with 2 attackers in Random attack.

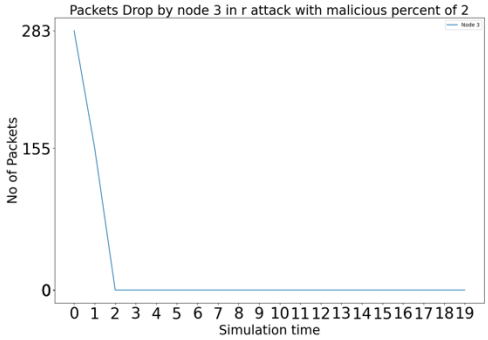

Fig 4.1.2.R.D.3: Packets Dropped by Node 3 with 2 attackers in Random attack.

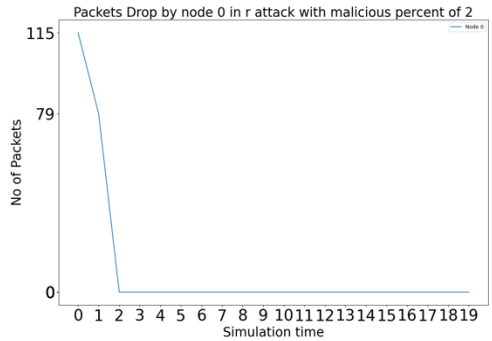

Fig 4.1.2.R.D.0: Packets Dropped by Node 0 with 2 attackers in Random attack.

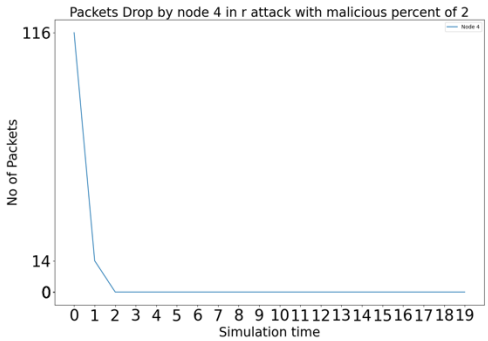

Fig 4.1.2.R.D.4: Packets Dropped by Node 4 with 2 attackers in Random attack.

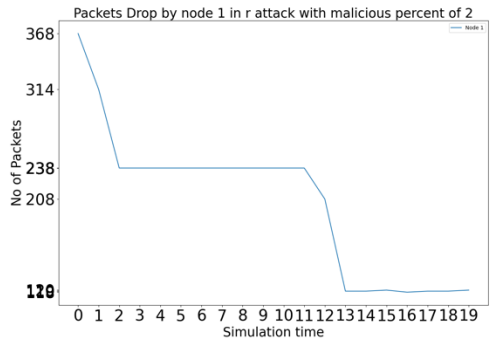

Fig 4.1.2.R.D.1: Packets Dropped by Node 1 with 2 attackers in Random attack.

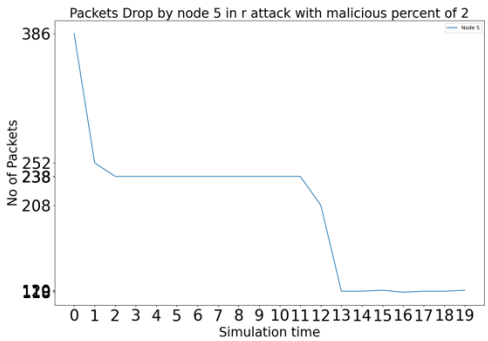

Fig 4.1.2.R.D.5: Packets Dropped by Node 5 with 2 attackers in Random attack.

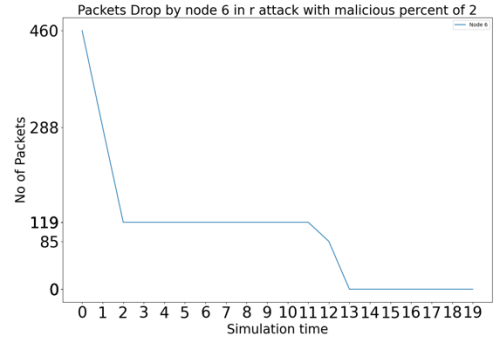

Fig 4.1.2.R.D.6: Packets Dropped by Node 6 with 2 attackers in Random attack.

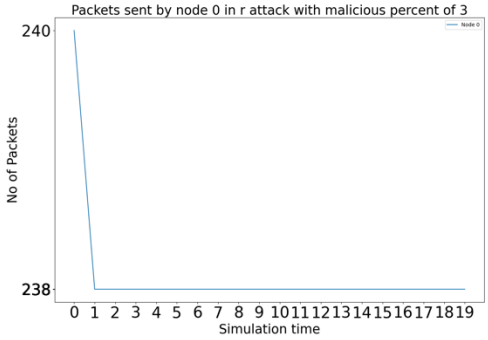

Fig 4.1.3.R.S.0: Packets Sent by Node 0 with 3 attackers in Random attack.

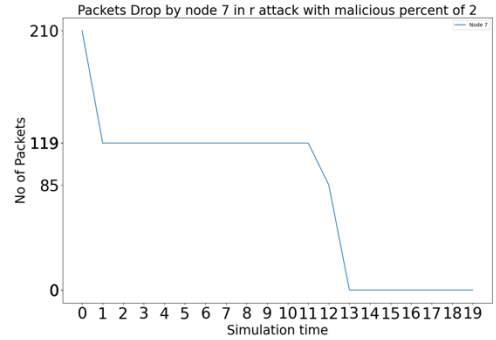

Fig 4.1.2.R.D.7: Packets Dropped by Node 7 with 2 attackers in Random attack.

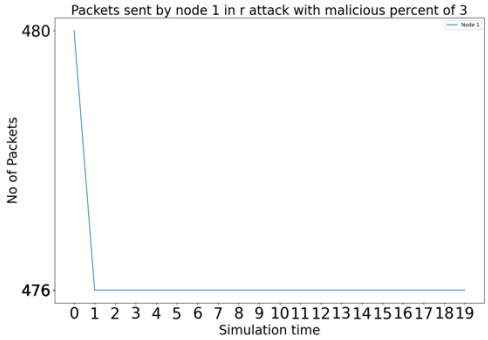

Fig 4.1.3.R.S.1: Packets Sent by Node 1 with 3 attackers in Random attack.

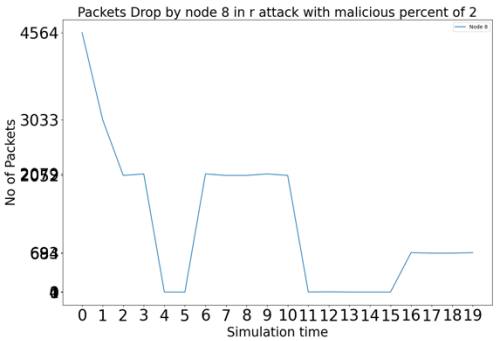

Fig 4.1.2.R.D.8: Packets Dropped by Node 8 with 2 attackers in Random attack.

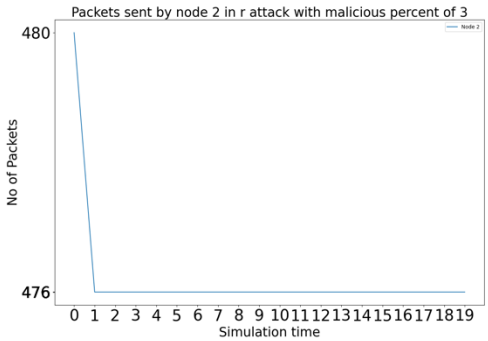

Fig 4.1.3.R.S.2: Packets Sent by Node 2 with 3 attackers in Random attack.

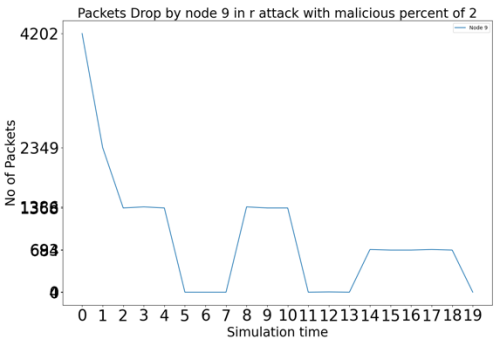

Fig 4.1.2.R.D.9: Packets Dropped by Node 9 with 2 attackers in Random attack.

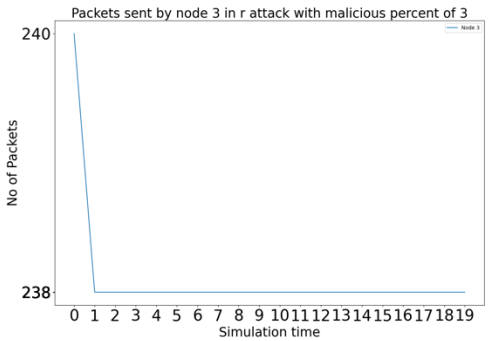

Fig 4.1.3.R.S.3: Packets Sent by Node 3 with 3 attackers in Random attack.

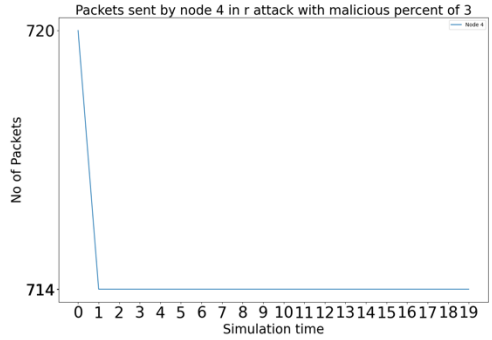

Fig 4.1.3.R.S.4: Packets Sent by Node 4 with 3 attackers in Random attack.

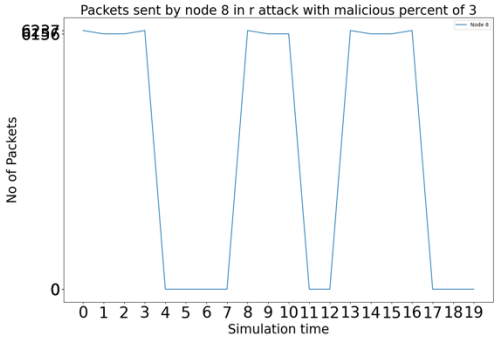

Fig 4.1.3.R.S.8: Packets Sent by Node 8 with 3 attackers in Random attack.

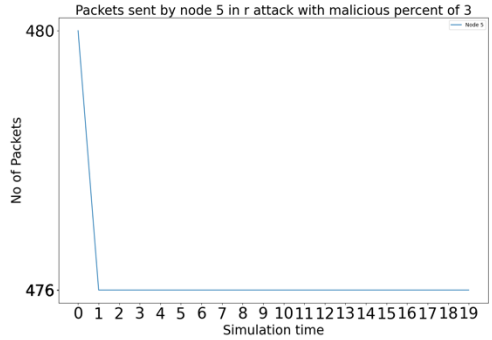

Fig 4.1.3.R.S.5: Packets Sent by Node 5 with 3 attackers in Random attack.

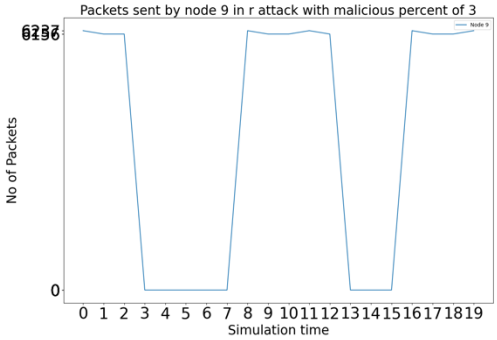

Fig 4.1.3.R.S.9: Packets Sent by Node 9 with 3 attackers in Random attack.

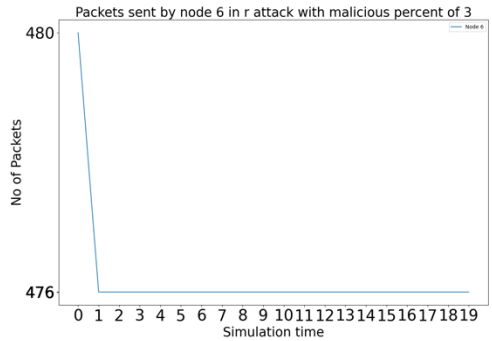

Fig 4.1.3.R.S.6: Packets Sent by Node 6 with 3 attackers in Random attack.

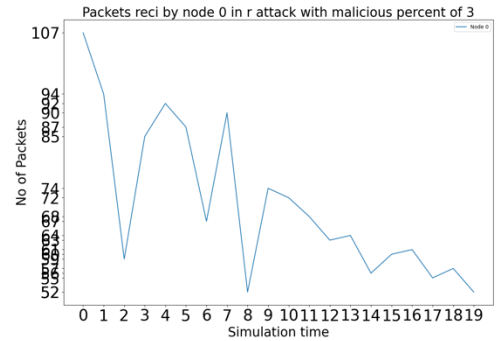

Fig 4.1.3.R.R.0: Packets Received by Node 0 with 3 attackers in Random attack.

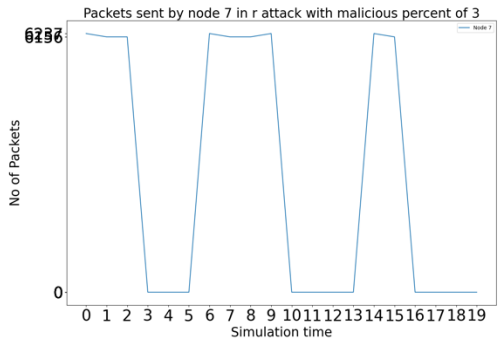

Fig 4.1.3.R.S.7: Packets Sent by Node 7 with 3 attackers in Random attack.

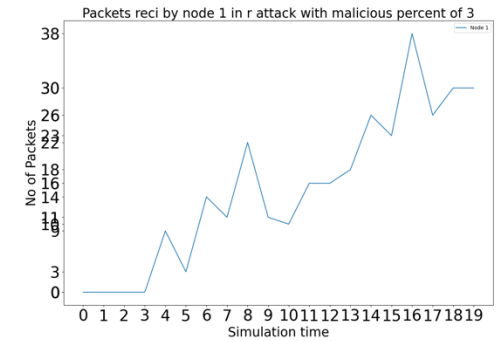

Fig 4.1.3.R.R.1: Packets Received by Node 1 with 3 attackers in Random attack.

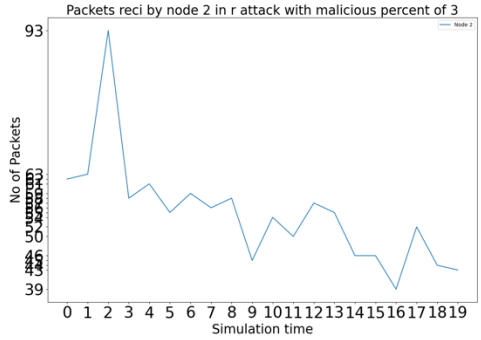

Fig 4.1.3.R.R.2: Packets Received by Node 2 with 3 attackers in Random attack.

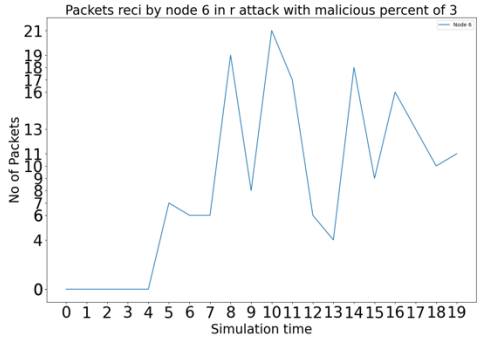

Fig 4.1.3.R.R.6: Packets Received by Node 6 with 3 attackers in Random attack.

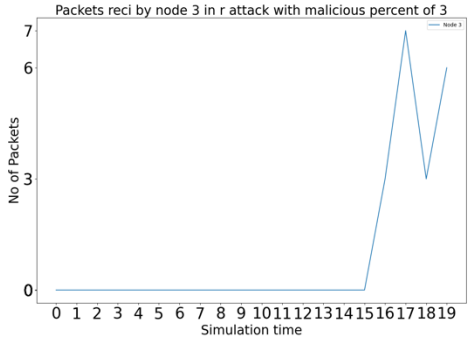

Fig 4.1.3.R.R.3: Packets Received by Node 3 with 3 attackers in Random attack.

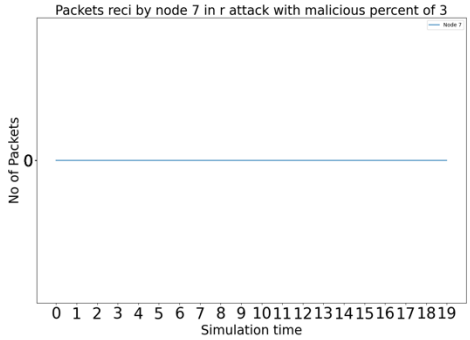

Fig 4.1.3.R.R.7: Packets Received by Node 7 with 3 attackers in Random attack.

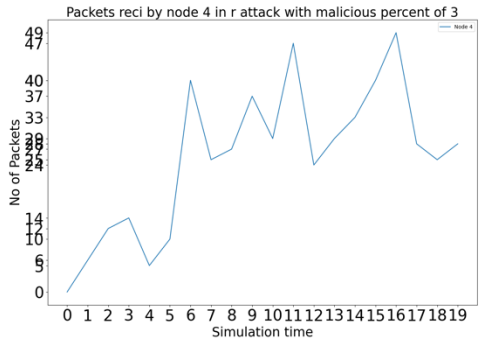

Fig 4.1.3.R.R.4: Packets Received by Node 4 with 3 attackers in Random attack.

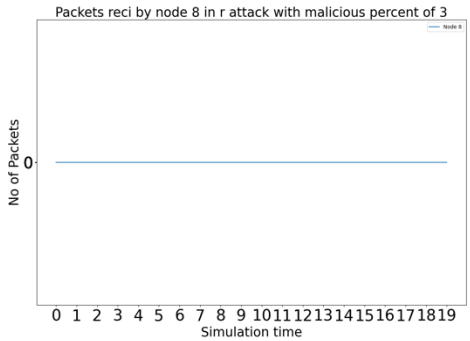

Fig 4.1.3.R.R.8: Packets Received by Node 8 with 3 attackers in Random attack.

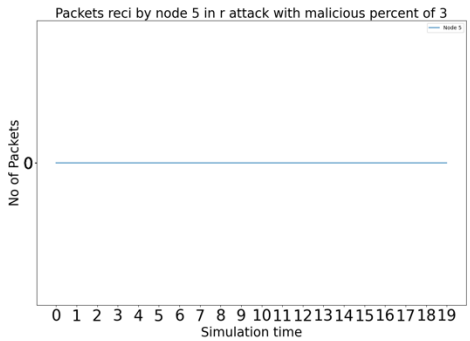

Fig 4.1.3.R.R.5: Packets Received by Node 5 with 3 attackers in Random attack.

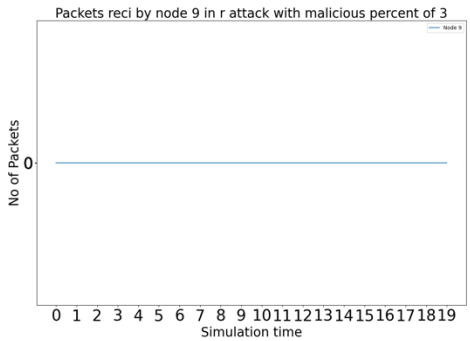

Fig 4.1.3.R.R.9: Packets Received by Node 9 with 3 attackers in Random attack.

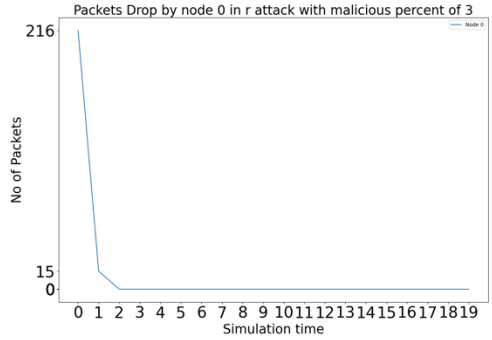

Fig 4.1.3.R.D.0: Packets Dropped by Node 0 with 3 attackers in Random attack.

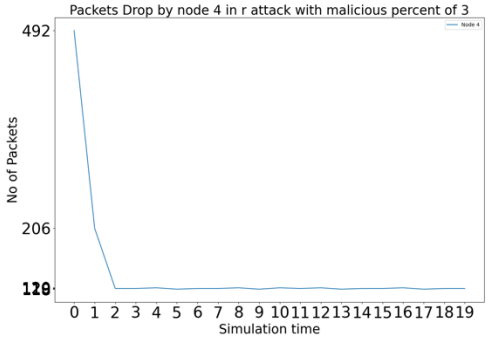

Fig 4.1.3.R.D.4: Packets Dropped by Node 4 with 3 attackers in Random attack.

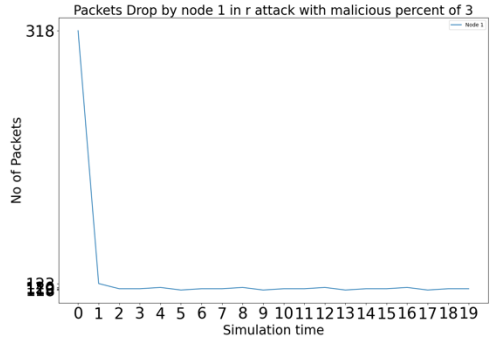

Fig 4.1.3.R.D.1: Packets Dropped by Node 1 with 3 attackers in Random attack.

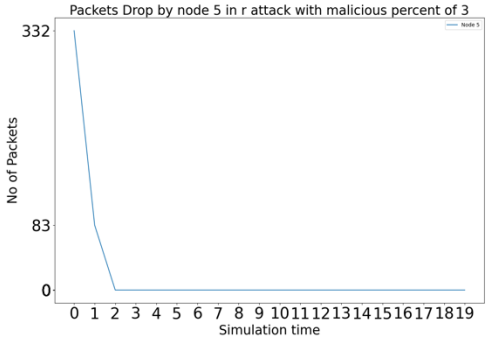

Fig 4.1.3.R.D.5: Packets Dropped by Node 5 with 3 attackers in Random attack.

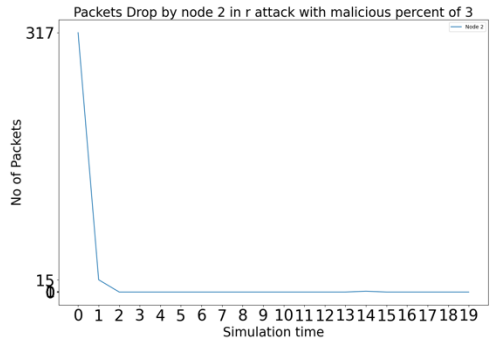

Fig 4.1.3.R.D.2: Packets Dropped by Node 2 with 3 attackers in Random attack.

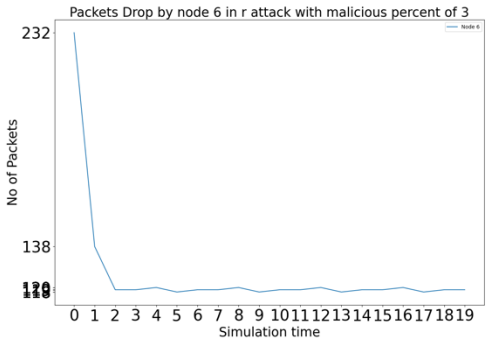

Fig 4.1.3.R.D.6: Packets Dropped by Node 6 with 3 attackers in Random attack.

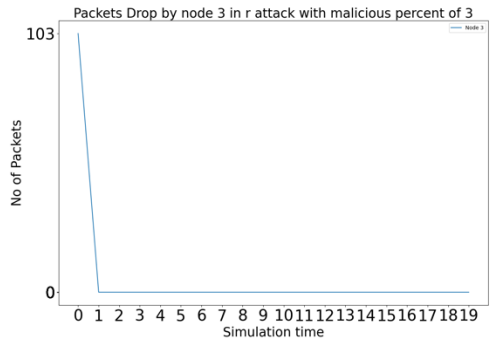

Fig 4.1.3.R.D.3: Packets Dropped by Node 3 with 3 attackers in Random attack.

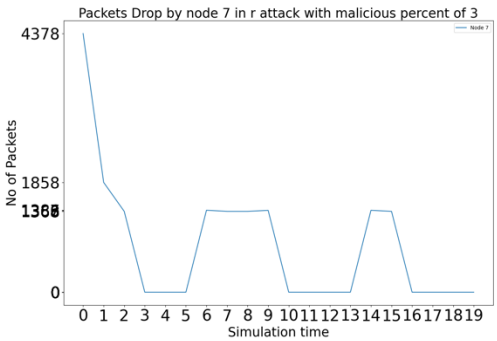

Fig 4.1.3.R.D.7: Packets Dropped by Node 7 with 3 attackers in Random attack.

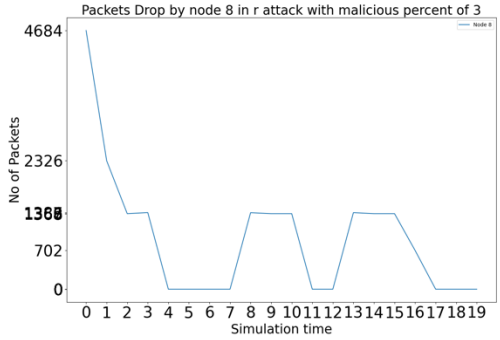

Fig 4.1.3.R.D.8: Packets Dropped by Node 8 with 3 attackers in Random attack.

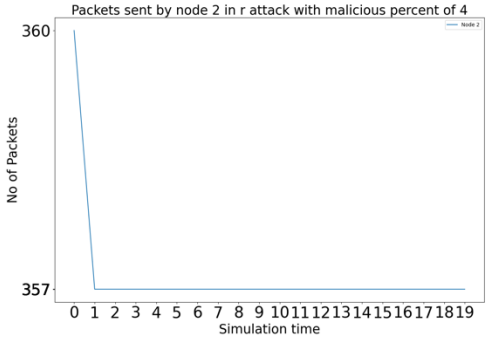

Fig 4.1.4.R.S.2: Packets Sent by Node 2 with 4 attackers in Random attack.

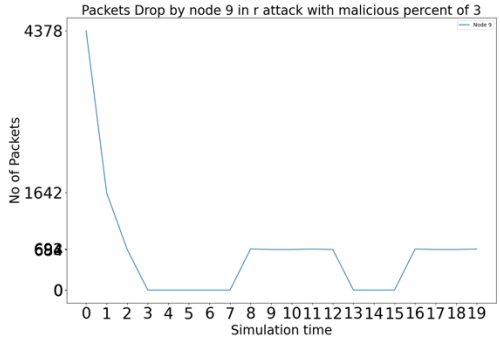

Fig 4.1.3.R.D.9: Packets Dropped by Node 9 with 3 attackers in Random attack.

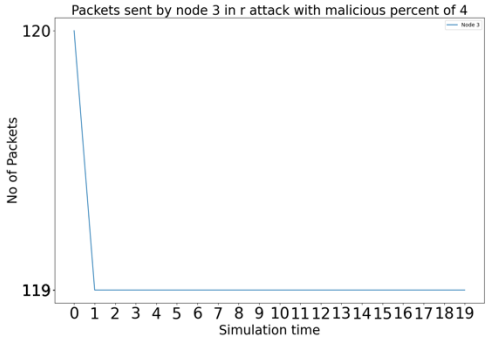

Fig 4.1.4.R.S.3: Packets Sent by Node 3 with 4 attackers in Random attack.

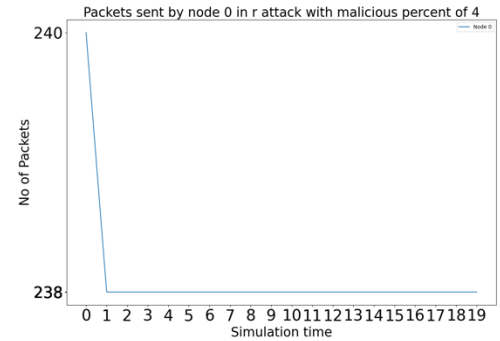

Fig 4.1.4.R.S.0: Packets Sent by Node 0 with 4 attackers in Random attack.

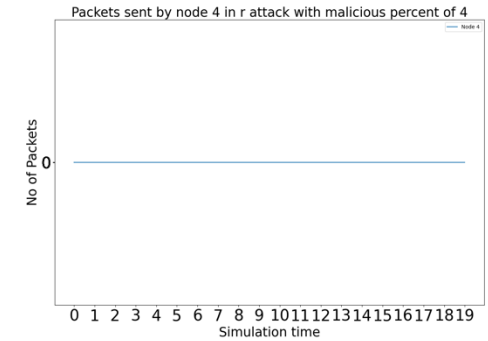

Fig 4.1.4.R.S.4: Packets Sent by Node 4 with 4 attackers in Random attack.

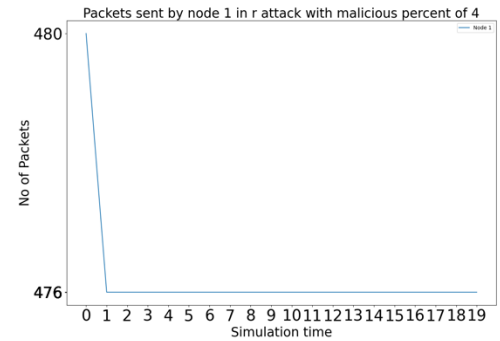

Fig 4.1.4.R.S.1: Packets Sent by Node 1 with 4 attackers in Random attack.

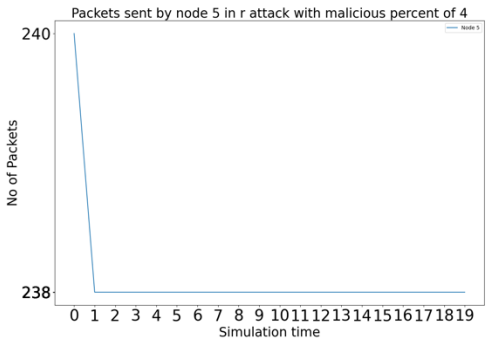

Fig 4.1.4.R.S.5: Packets Sent by Node 5 with 4 attackers in Random attack.

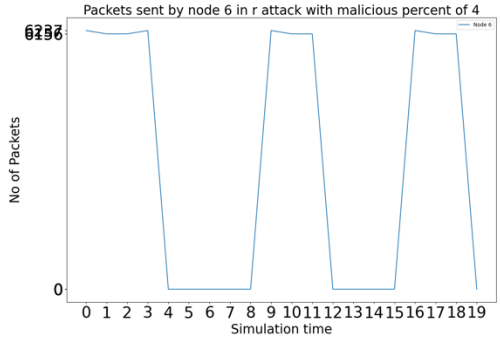

Fig 4.1.4.R.S.6: Packets Sent by Node 6 with 4 attackers in Random attack.

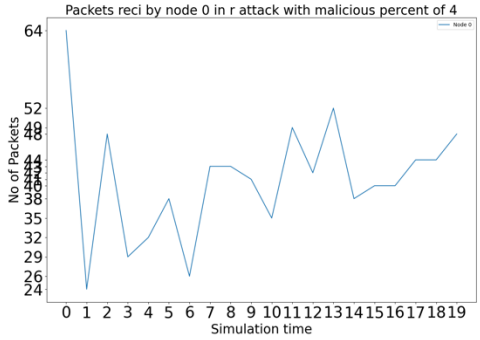

Fig 4.1.4.R.R.0: Packets Received by Node 0 with 4 attackers in Random attack.

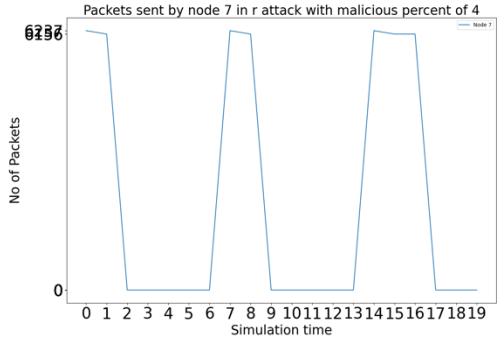

Fig 4.1.4.R.S.7: Packets Sent by Node 7 with 4 attackers in Random attack.

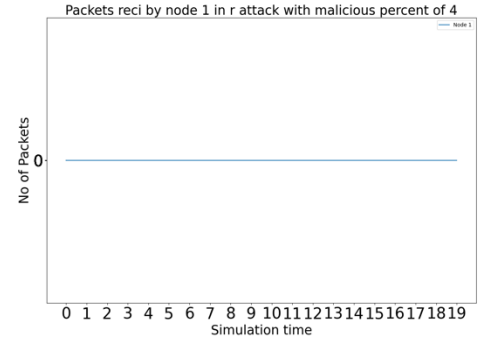

Fig 4.1.4.R.R.1: Packets Received by Node 1 with 4 attackers in Random attack.

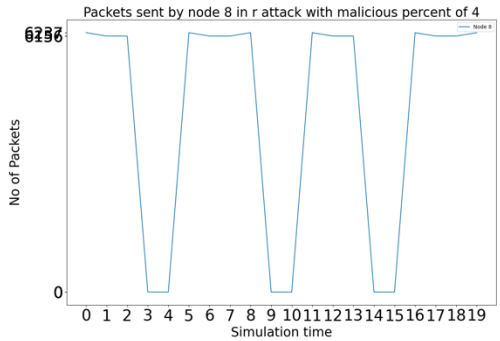

Fig 4.1.4.R.S.8: Packets Sent by Node 8 with 4 attackers in Random attack.

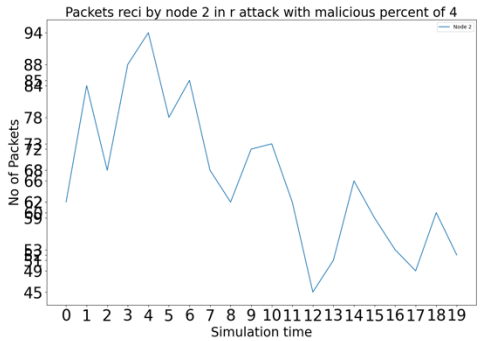

Fig 4.1.4.R.R.2: Packets Received by Node 2 with 4 attackers in Random attack.

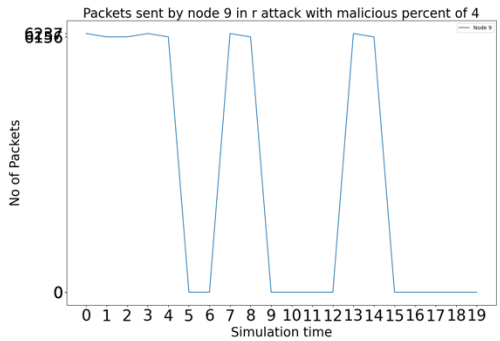

Fig 4.1.4.R.S.9: Packets Sent by Node 9 with 4 attackers in Random attack.

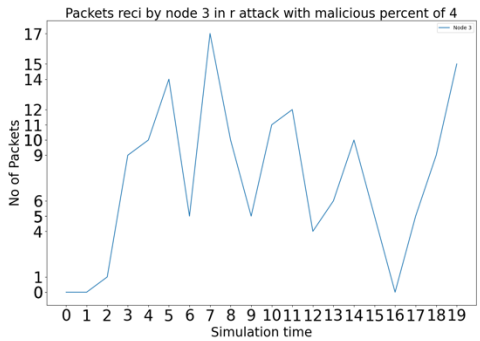

Fig 4.1.4.R.R.3: Packets Received by Node 3 with 4 attackers in Random attack.

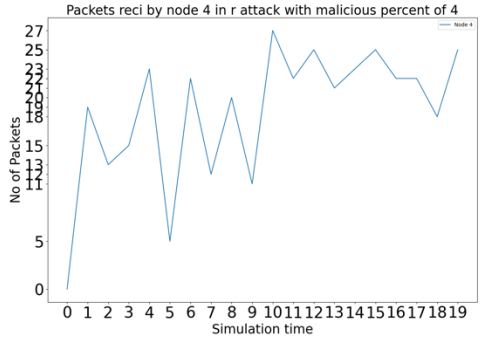

Fig 4.1.4.R.R.4: Packets Received by Node 4 with 4 attackers in Random attack.

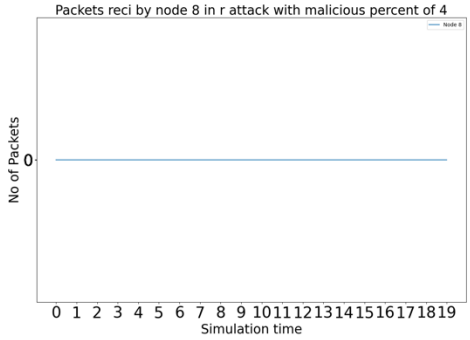

Fig 4.1.4.R.R.8: Packets Received by Node 8 with 4 attackers in Random attack.

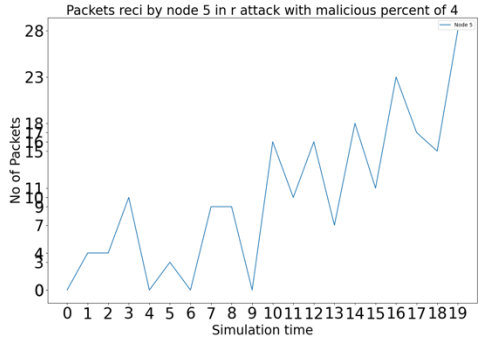

Fig 4.1.4.R.R.5: Packets Received by Node 5 with 4 attackers in Random attack.

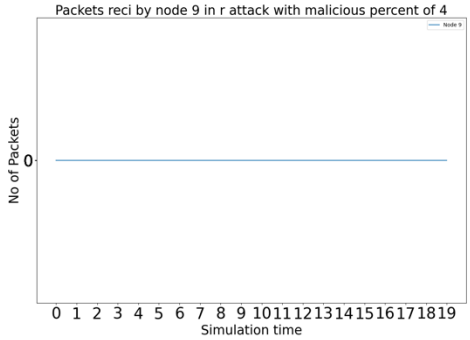

Fig 4.1.4.R.R.9: Packets Received by Node 9 with 4 attackers in Random attack.

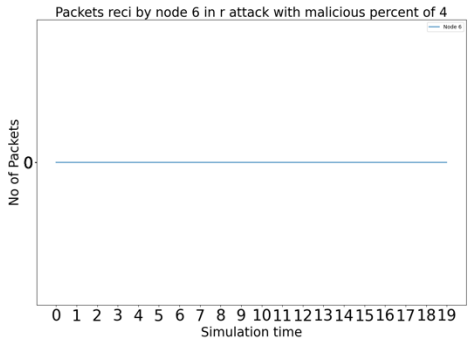

Fig 4.1.4.R.R.6: Packets Received by Node 6 with 4 attackers in Random attack.

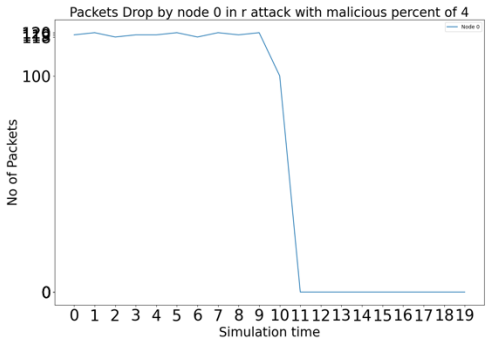

Fig 4.1.4.R.D.0: Packets Dropped by Node 0 with 4 attackers in Random attack.

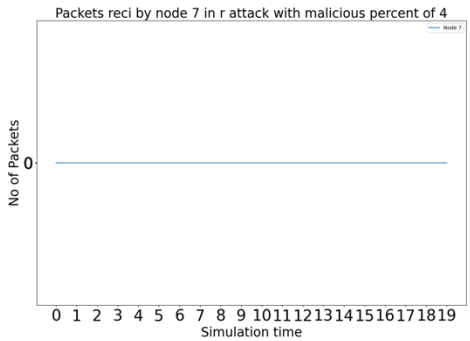

Fig 4.1.4.R.R.7: Packets Received by Node 7 with 4 attackers in Random attack.

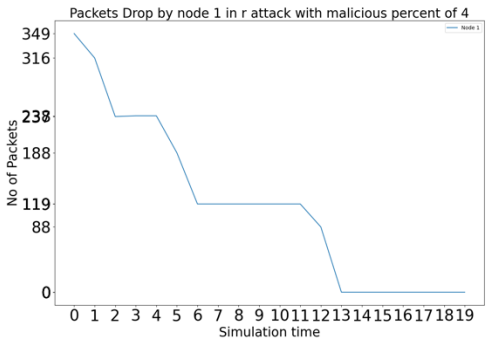

Fig 4.1.4.R.D.1: Packets Dropped by Node 1 with 4 attackers in Random attack.

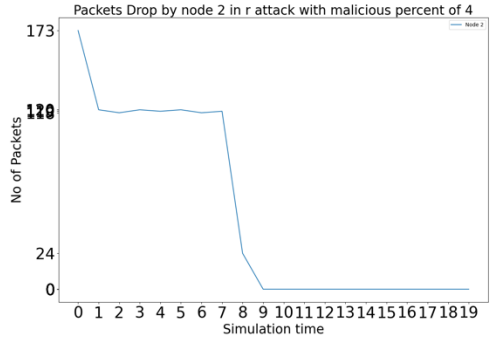

Fig 4.1.4.R.D.2: Packets Dropped by Node 2 with 4 attackers in Random attack.

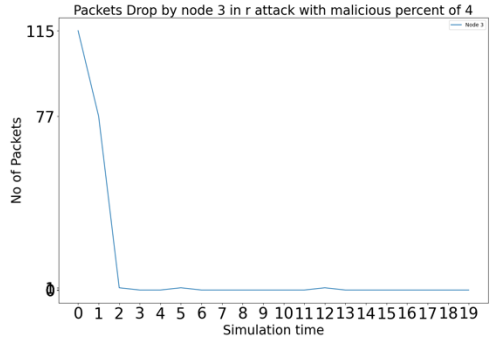

Fig 4.1.4.R.D.3: Packets Dropped by Node 3 with 4 attackers in Random attack.

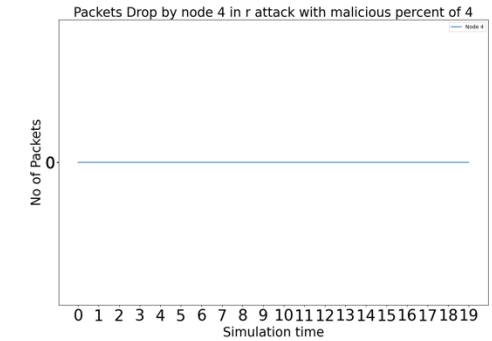

Fig 4.1.4.R.D.4: Packets Dropped by Node 4 with 4 attackers in Random attack.

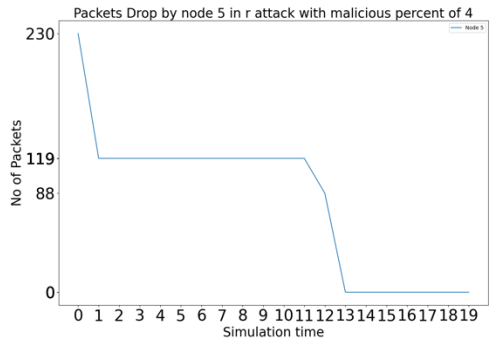

Fig 4.1.4.R.D.5: Packets Dropped by Node 5 with 4 attackers in Random attack.

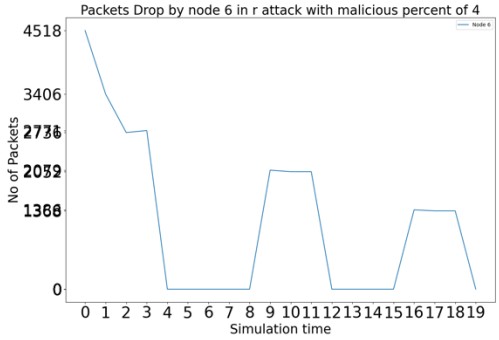

Fig 4.1.4.R.D.6: Packets Dropped by Node 6 with 4 attackers in Random attack.

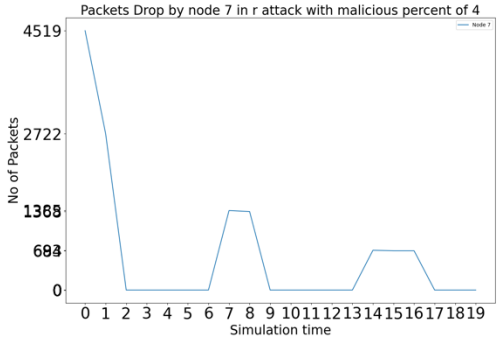

Fig 4.1.4.R.D.7: Packets Dropped by Node 7 with 4 attackers in Random attack.

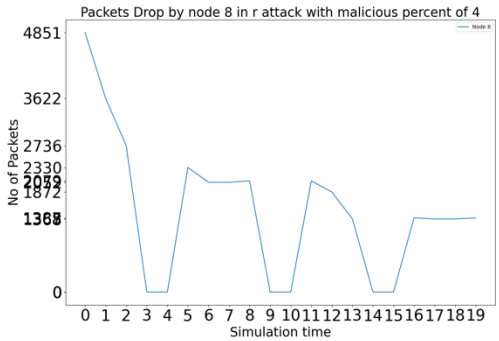

Fig 4.1.4.R.D.8: Packets Dropped by Node 8 with 4 attackers in Random attack.

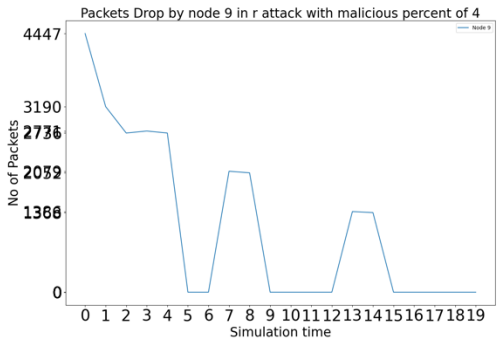

Fig 4.1.4.R.D.9: Packets Dropped by Node 9 with 4 attackers in Random attack.

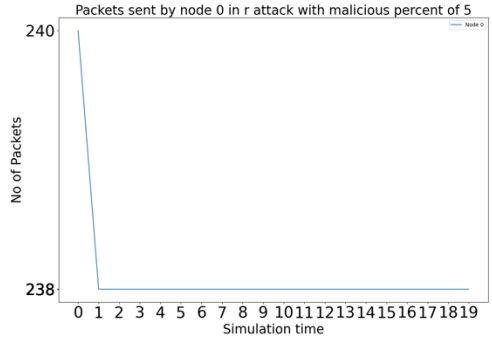

Fig 4.1.5.R.S.0: Packets Sent by Node 0 with 5 attackers in Random attack.

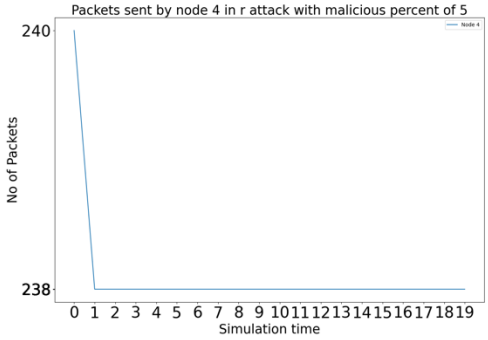

Fig 4.1.5.R.S.4: Packets Sent by Node 4 with 5 attackers in Random attack.

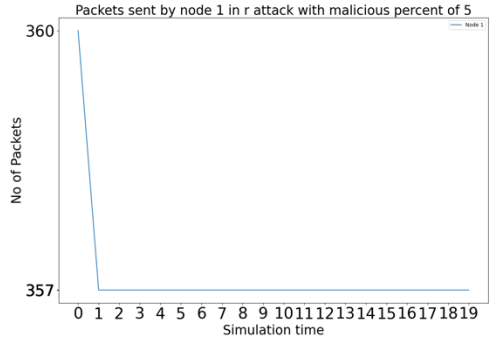

Fig 4.1.5.R.S.1: Packets Sent by Node 1 with 5 attackers in Random attack.

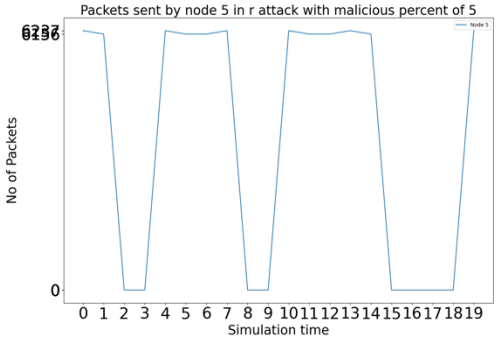

Fig 4.1.5.R.S.5: Packets Sent by Node 5 with 5 attackers in Random attack.

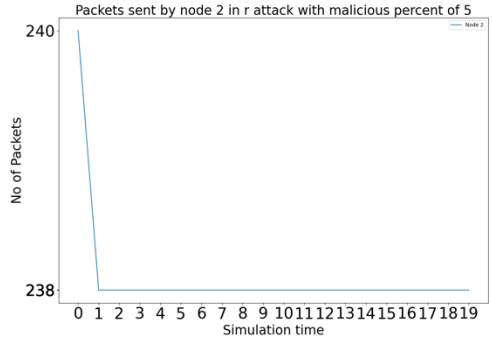

Fig 4.1.5.R.S.2: Packets Sent by Node 2 with 5 attackers in Random attack.

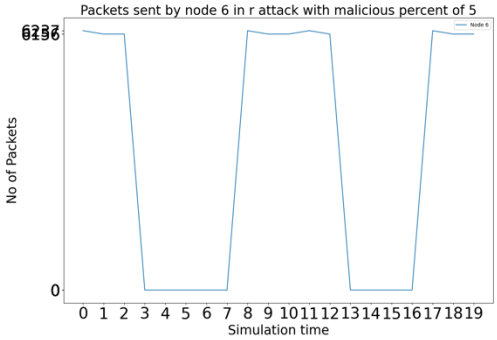

Fig 4.1.5.R.S.6: Packets Sent by Node 6 with 5 attackers in Random attack.

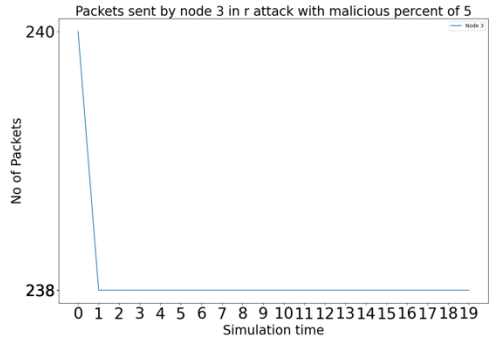

Fig 4.1.5.R.S.3: Packets Sent by Node 3 with 5 attackers in Random attack.

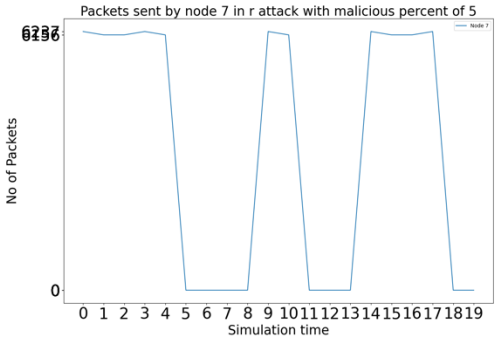

Fig 4.1.5.R.S.7: Packets Sent by Node 7 with 5 attackers in Random attack.

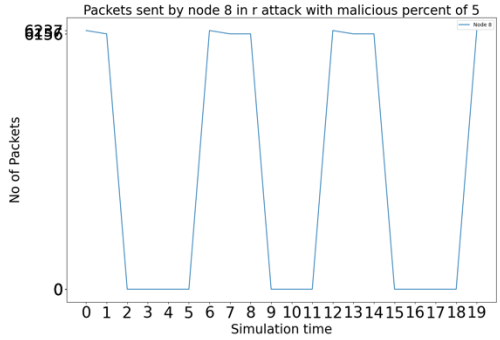

Fig 4.1.5.R.S.8: Packets Sent by Node 8 with 5 attackers in Random attack.

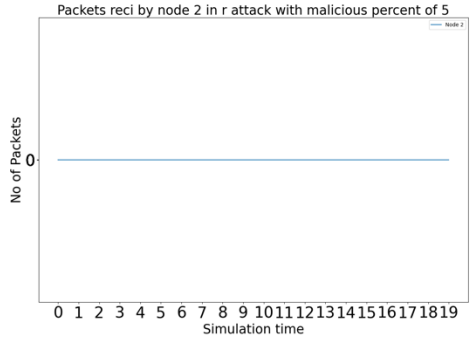

Fig 4.1.5.R.R.2: Packets Received by Node 2 with 5 attackers in Random attack.

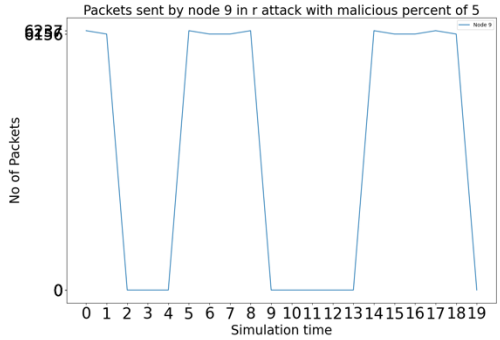

Fig 4.1.5.R.S.9: Packets Sent by Node 9 with 5 attackers in Random attack.

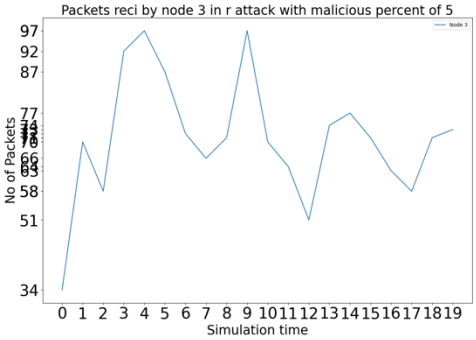

Fig 4.1.5.R.R.3: Packets Received by Node 3 with 5 attackers in Random attack.

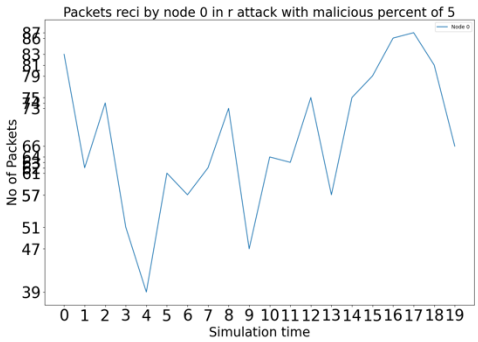

Fig 4.1.5.R.R.0: Packets Received by Node 0 with 5 attackers in Random attack.

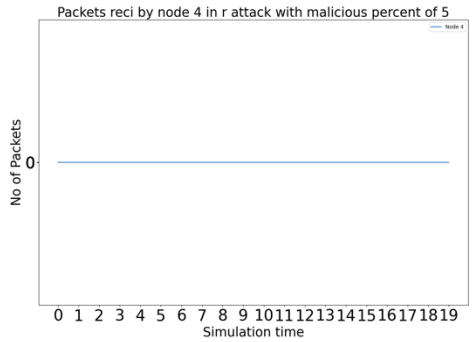

Fig 4.1.5.R.R.4: Packets Received by Node 4 with 5 attackers in Random attack.

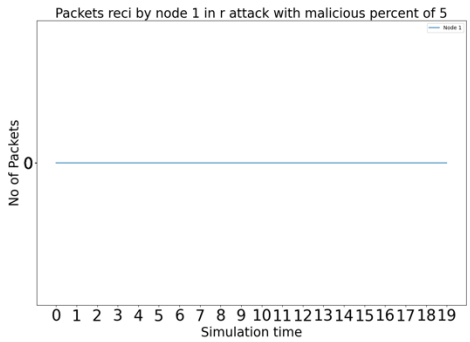

Fig 4.1.5.R.R.1: Packets Received by Node 1 with 5 attackers in Random attack.

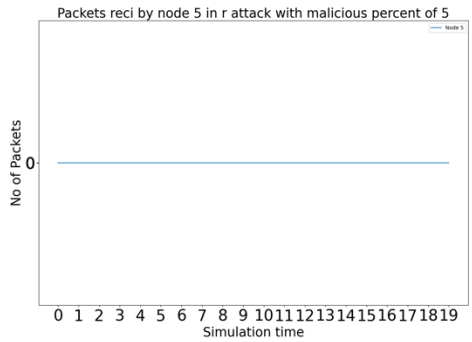

Fig 4.1.5.R.R.5: Packets Received by Node 5 with 5 attackers in Random attack.

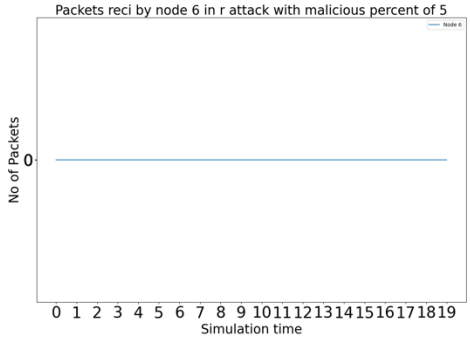

Fig 4.1.5.R.R.6: Packets Received by Node 6 with 5 attackers in Random attack.

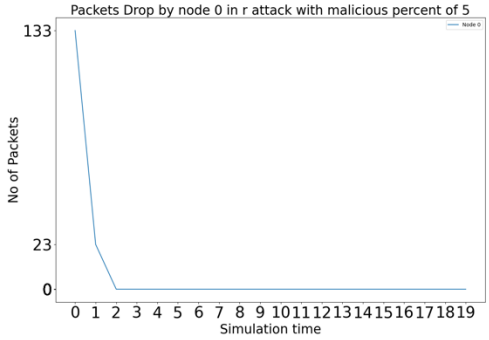

Fig 4.1.5.R.D.0: Packets Dropped by Node 0 with 5 attackers in Random attack.

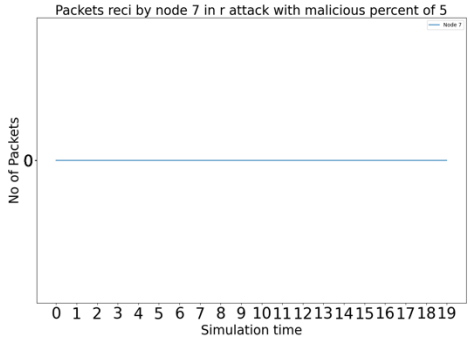

Fig 4.1.5.R.R.7: Packets Received by Node 7 with 5 attackers in Random attack.

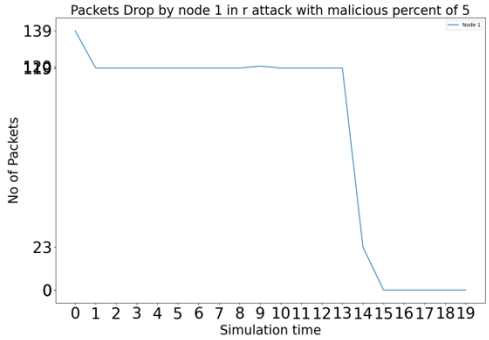

Fig 4.1.5.R.D.1: Packets Dropped by Node 1 with 5 attackers in Random attack.

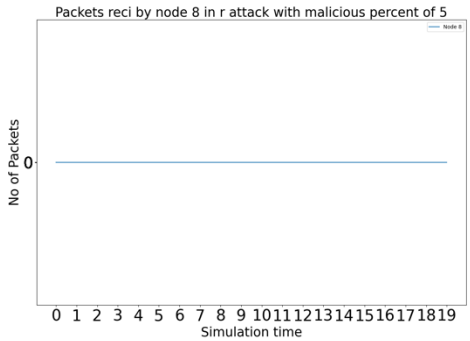

Fig 4.1.5.R.R.8: Packets Received by Node 8 with 5 attackers in Random attack.

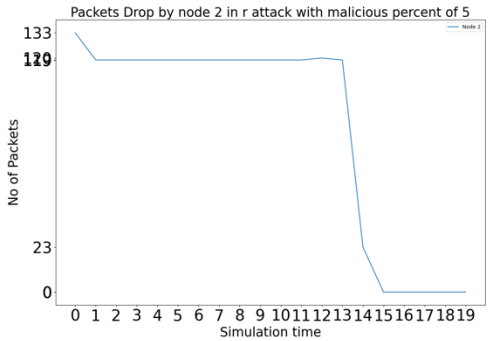

Fig 4.1.5.R.D.2: Packets Dropped by Node 2 with 5 attackers in Random attack.

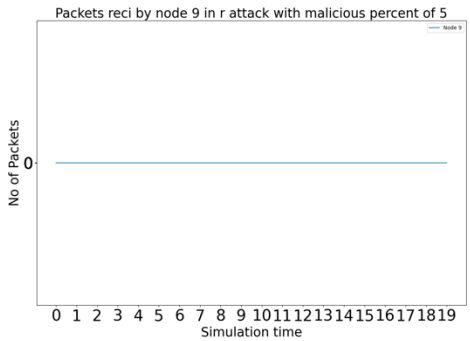

Fig 4.1.5.R.R.9: Packets Received by Node 9 with 5 attackers in Random attack.

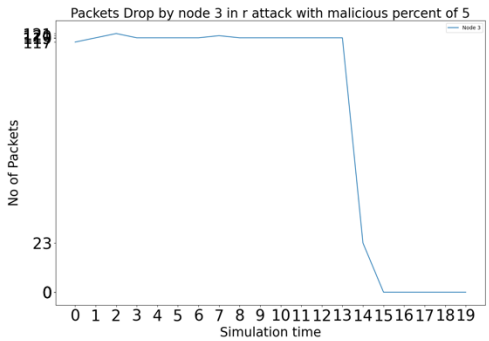

Fig 4.1.5.R.D.3: Packets Dropped by Node 3 with 5 attackers in Random attack.

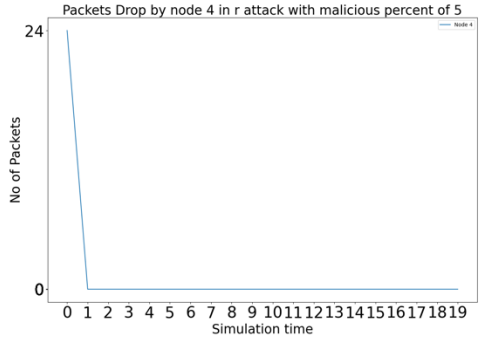

Fig 4.1.5.R.D.4: Packets Dropped by Node 4 with 5 attackers in Random attack.

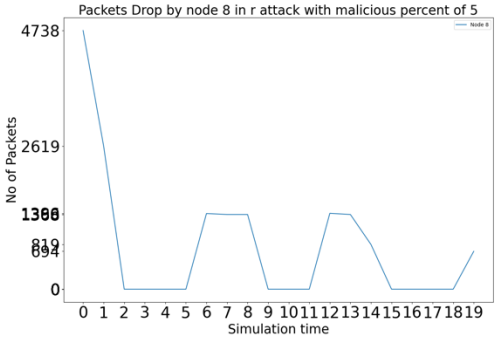

Fig 4.1.5.R.D.8: Packets Dropped by Node 8 with 5 attackers in Random attack.

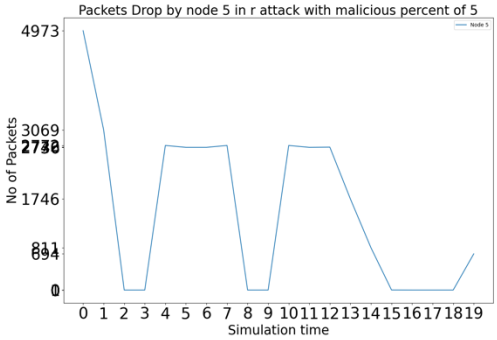

Fig 4.1.5.R.D.5: Packets Dropped by Node 5 with 5 attackers in Random attack.

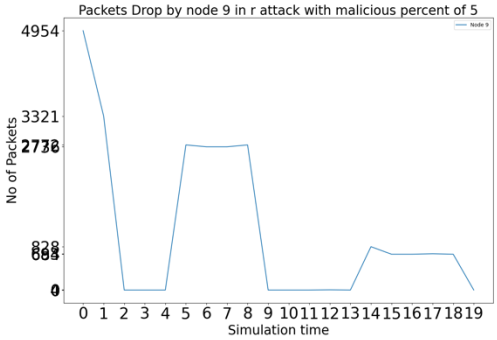

Fig 4.1.5.R.D.9: Packets Dropped by Node 9 with 5 attackers in Random attack.

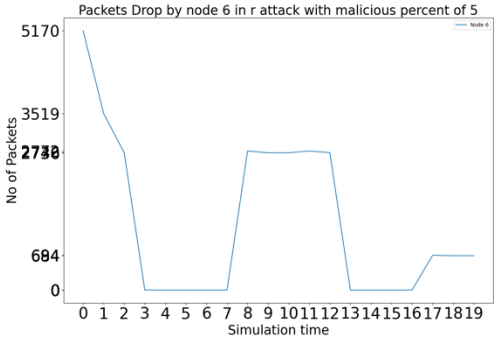

Fig 4.1.5.R.D.6: Packets Dropped by Node 6 with 5 attackers in Random attack.

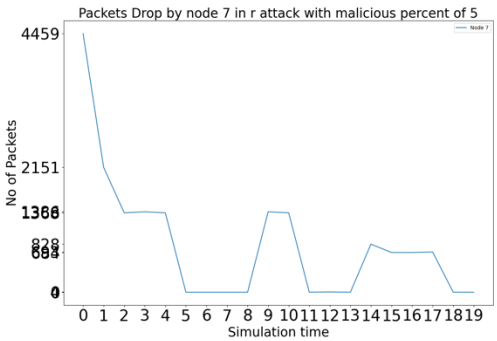

Fig 4.1.5.R.D.7: Packets Dropped by Node 7 with 5 attackers in Random attack.

4.2 Results of Packet Delivery Ratio and Packet Drop Ratio

Constant Attack: Packet Delivery Ratio

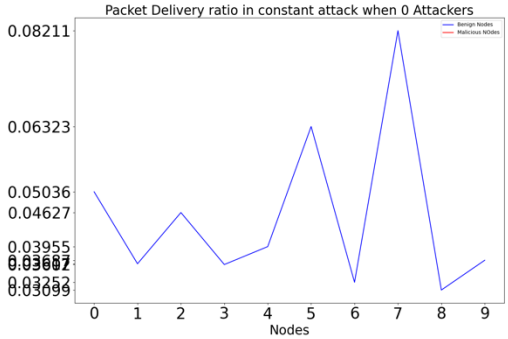

Fig 4.2.C.1.0: Packet Delivery Ratio of the network with 0 attackers in Constant attack.

Fig 4.2.C.1.3: Packet Delivery Ratio of the network with 3 attackers in Constant attack.

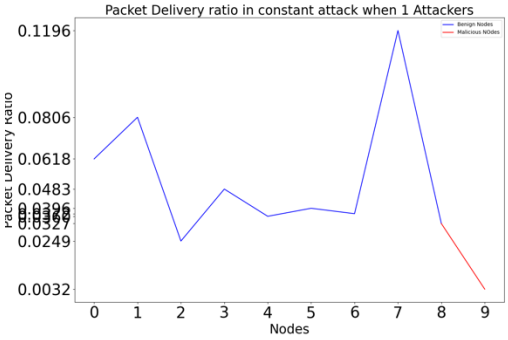

Fig 4.2.C.1.1: Packet Delivery Ratio of the network with 1 attacker in Constant attack.

Fig 4.2.C.1.4: Packet Delivery Ratio of the network with 4 attackers in Constant attack.

Fig 4.2.C.1.2: Packet Delivery Ratio of the network with 2 attackers in Constant attack.

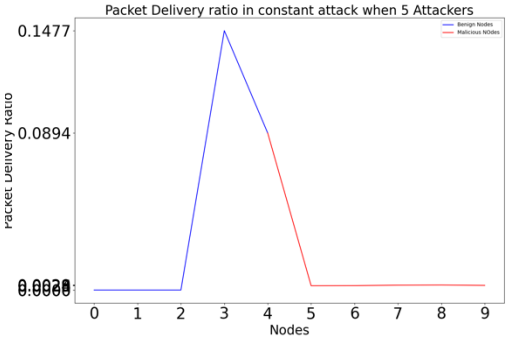

Fig 4.2.C.1.5: Packet Delivery Ratio of the network with 5 attackers in Constant attack.

Constant Attack: Packet Drop Ratio

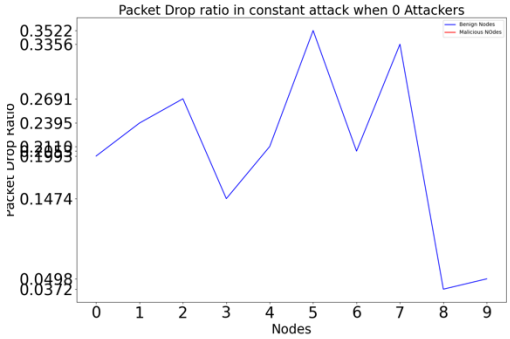

Fig 4.2.C.2.0: Packet Drop Ratio of the network with 0 attackers in Constant attack.

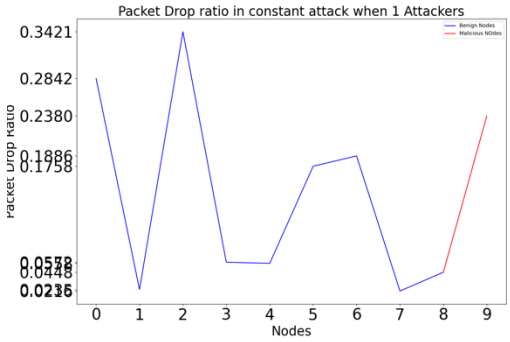

Fig 4.2.C.2.1: Packet Drop Ratio of the network with 1 attacker in Constant attack.

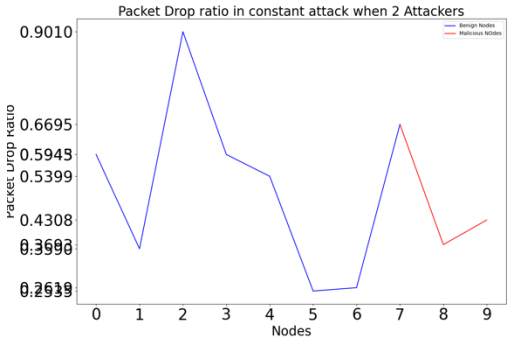

Fig 4.2.C.2.2: Packet Drop Ratio of the network with 2 attackers in Constant attack.

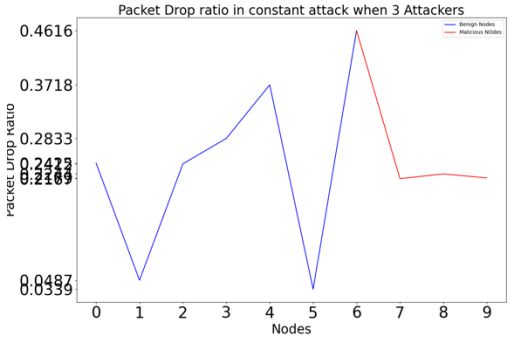

Fig 4.2.C.2.3: Packet Drop Ratio of the network with 3 attackers in Constant attack.

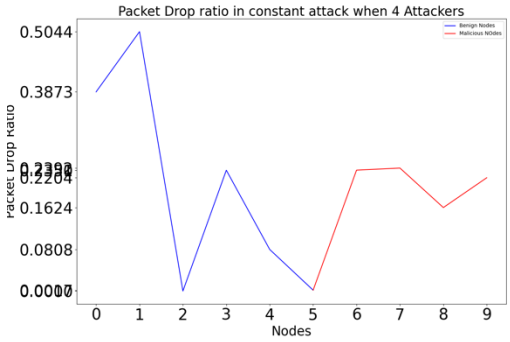

Fig 4.2.C.2.4: Packet Drop Ratio of the network with 4 attackers in Constant attack.

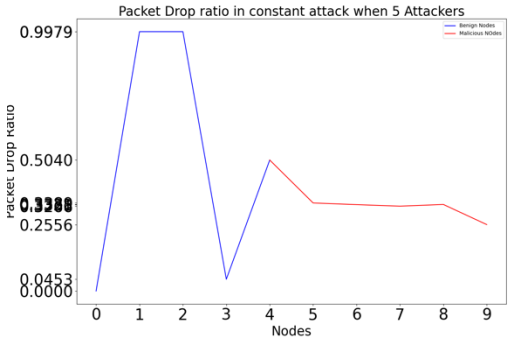

Fig 4.2.C.2.5: Packet Drop Ratio of the network with 5 attackers in Constant attack.

Periodic Attack: Packet Delivery Ratio

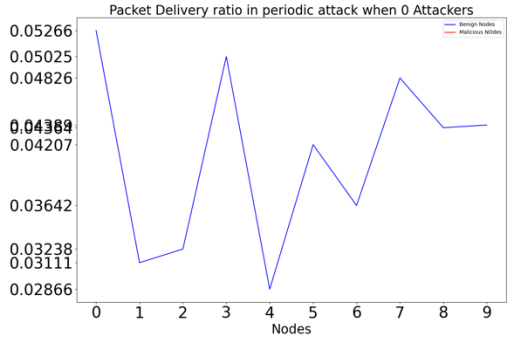

Fig 4.2.P.1.0: Packet Delivery Ratio of the network with 0 attackers in Periodic attack.

Fig 4.2.P.1.3: Packet Delivery Ratio of the network with 3 attackers in Periodic attack.

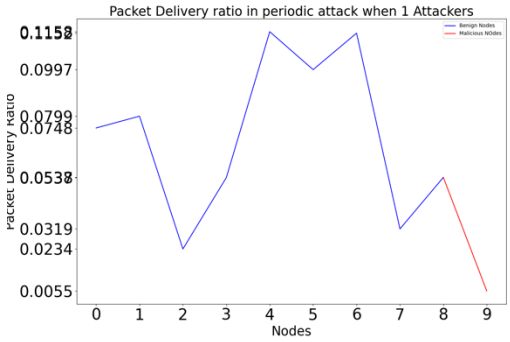

Fig 4.2.P.1.1: Packet Delivery Ratio of the network with 1 attacker in Periodic attack.

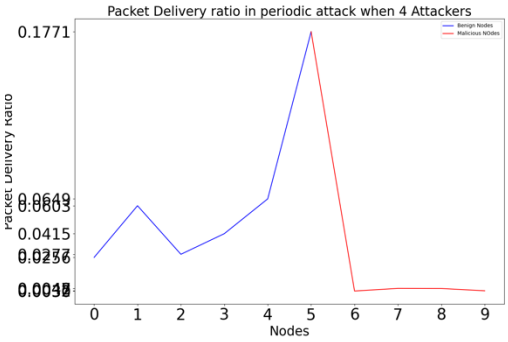

Fig 4.2.P.1.4: Packet Delivery Ratio of the network with 4 attackers in Periodic attack.

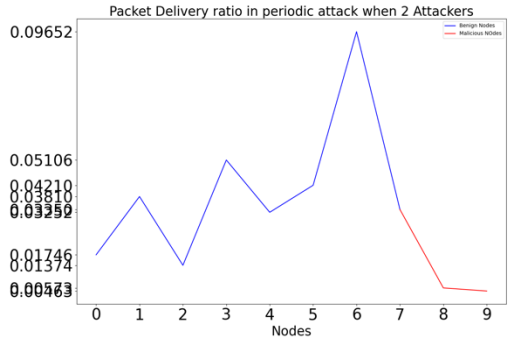

Fig 4.2.P.1.2: Packet Delivery Ratio of the network with 2 attackers in Periodic attack.

Fig 4.2.P.1.5: Packet Delivery Ratio of the network with 5 attackers in Periodic attack.

Periodic Attack: Packet Drop Ratio

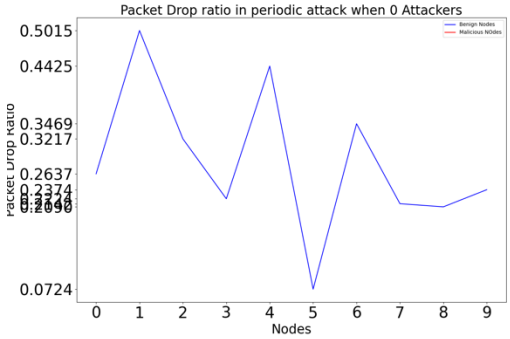

Fig 4.2.P.2.0: Packet Drop Ratio of the network with 0 attackers in Periodic attack.

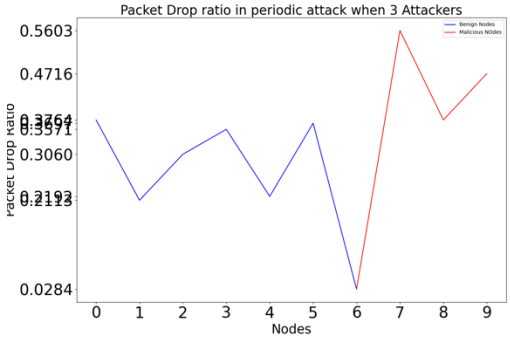

Fig 4.2.P.2.3: Packet Drop Ratio of the network with 3 attackers in Periodic attack.

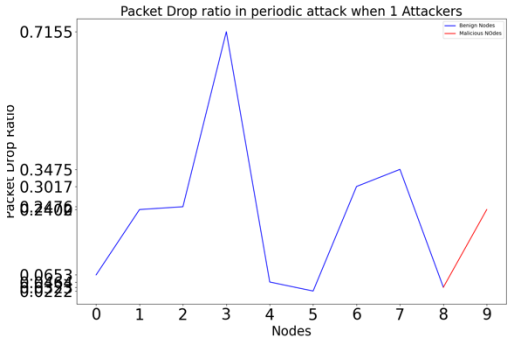

Fig 4.2.P.2.1: Packet Drop Ratio of the network with 1 attacker in Periodic attack.

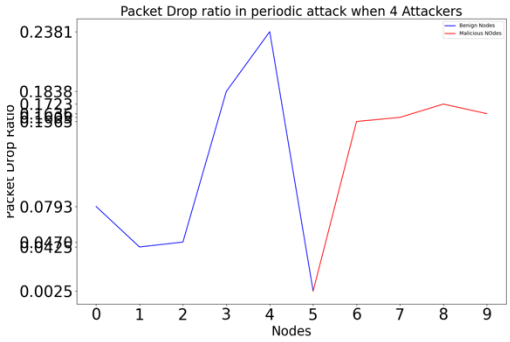

Fig 4.2.P.2.4: Packet Drop Ratio of the network with 4 attackers in Periodic attack.

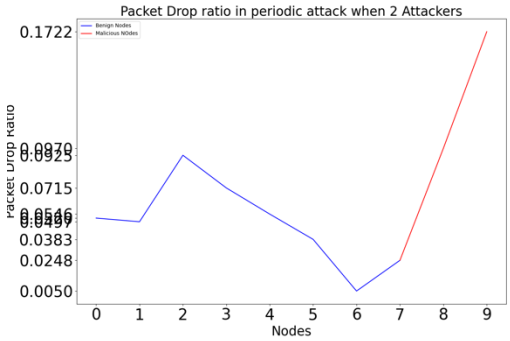

Fig 4.2.P.2.2: Packet Drop Ratio of the network with 2 attackers in Periodic attack.

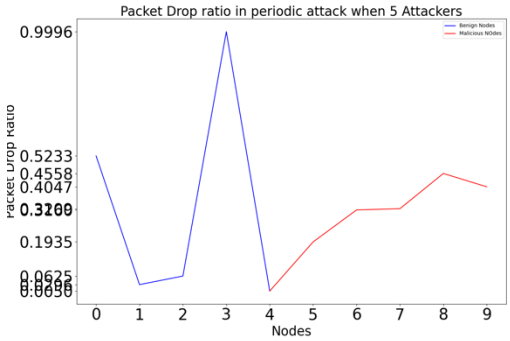

Fig 4.2.P.2.5: Packet Drop Ratio of the network with 5 attackers in Periodic attack.

Random Attack: Packet Delivery Ratio

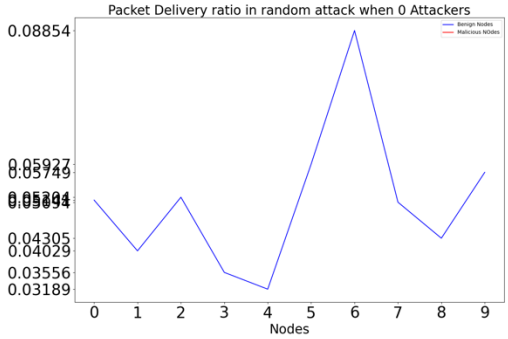

Fig 4.2.R.1.0: Packet Delivery Ratio of the network with 0 attackers in Random attack.

Fig 4.2.R.1.3: Packet Delivery Ratio of the network with 3 attackers in Random attack.

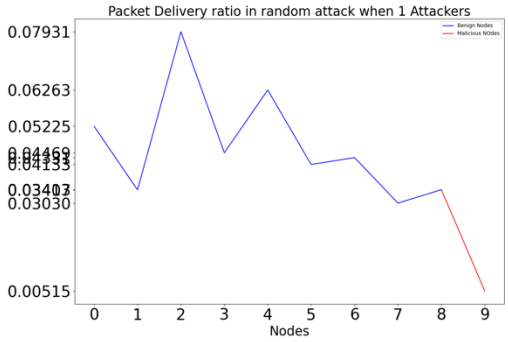

Fig 4.2.R.1.1: Packet Delivery Ratio of the network with 1 attacker in Random attack.

Fig 4.2.R.1.4: Packet Delivery Ratio of the network with 4 attackers in Random attack.

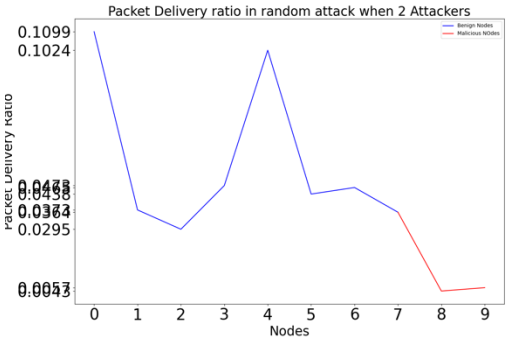

Fig 4.2.R.1.2: Packet Delivery Ratio of the network with 2 attackers in Random attack.

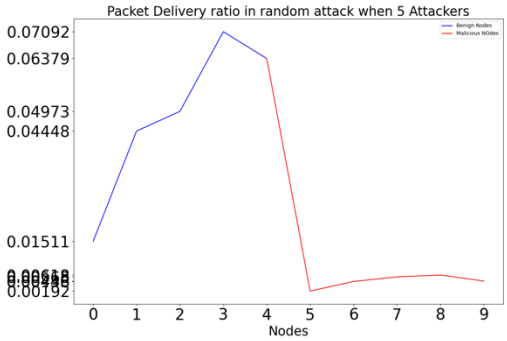

Fig 4.2.R.1.5: Packet Delivery Ratio of the network with 5 attackers in Random attack.

Random Attack: Packet Drop Ratio

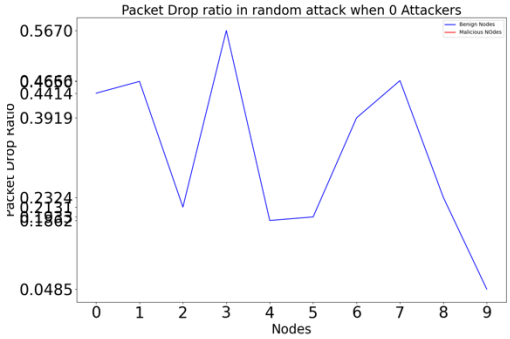

Fig 4.2.R.2.0: Packet Drop Ratio of the network with 0 attackers in Random attack.

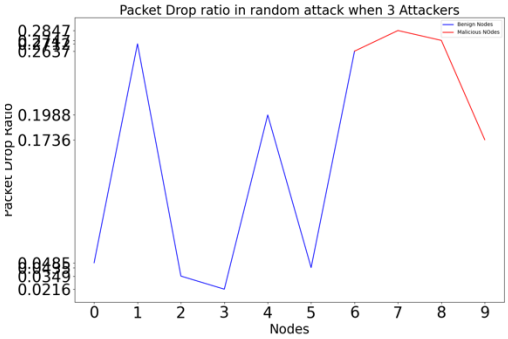

Fig 4.2.R.2.3: Packet Drop Ratio of the network with 3 attackers in Random attack.

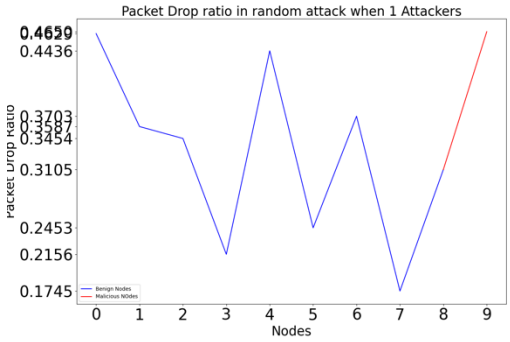

Fig 4.2.R.2.1: Packet Drop Ratio of the network with 1 attacker in Random attack.

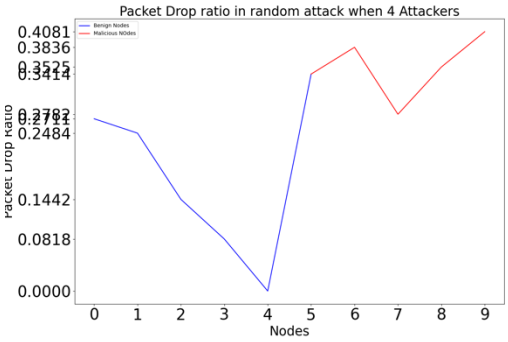

Fig 4.2.R.2.4: Packet Drop Ratio of the network with 4 attackers in Random attack.

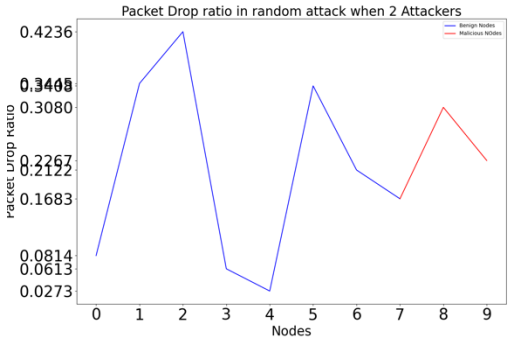

Fig 4.2.R.2.2: Packet Drop Ratio of the network with 2 attackers in Random attack.

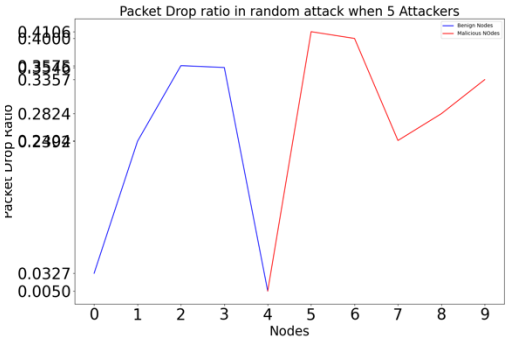

Fig 4.2.R.2.5: Packet Drop Ratio of the network with 5 attackers in Random attack.

CHAPTER – 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

The paper presented the detection mechanism to detect various jamming attack nodes in MANET. The network is created and simulated in NS2 tool with 10 nodes out of which all the nodes are behaving like normal nodes at one time later some of the nodes shows normal behavior and other nodes shows different jamming behavior like constant, periodic and random jammers varying from single attack node to multiple attack nodes. Then the like packet sent, received and dropped collected from trace file and analyzed using python scripts, from these characteristics the packets delivery ratio and packet drop ratio calculated and based on this behavior the malicious node was detected by this the network performance can be unproved by re-routing the network.

5.2 FUTURE SCOPE

In future,

1. The behavioral analysis of jamming nodes can also be classified using Machine Learning Algorithms.
2. Not only characteristics like number of packets sent, number of packets received and number of packets dropped there are many more characteristics for classifying the malicious nodes from normal nodes.
3. In this project all the malicious nodes and the normal nodes start at 0^{th} second this can be modified as malicious nodes starting from a particular second like 5^{th} or 10^{th} or any such. This helps while comparing the characteristics of normal network to a network with malicious nodes.

BIBILOGRAPHY

[1] Adilakshmi, Y. (2022). Node Behaviour Classification Using SVM & Decision Tree to Detect Malicious Nodes in MANET. The International Journal of Analytical and Experimental Modal Analysis, ISSN NO: 0886-9367.

[2] Indira, D.N.V.S.L.S., Abinaya, R., et al. (2021). Secured Personal Health Records using Pattern Based Verification and 2-Way Polynomial Protocol in Cloud Infrastructure. International Journal of Ad Hoc and Ubiquitous Computing, 40(3), 86-93. ISSN: 1743-8233.

[3] Kim, J., Biswas, P. K., Bohacek, S., Mackey, S. J., Samoohi, S., & Patel, M. P. (2021). Advanced protocols for the mitigation of friendly jamming in mobile ad-hoc networks. Journal of Network and Computer Applications, 181, 103037.

[4] S. Shrestha, R. Baidya, B. Giri, and A. Thapa, "Securing Blackhole Attacks in MANETs using Modified Sequence Number in AODV Routing Protocol," in 2020 8th International Electrical Engineering Congress (iEECON), Chiang Mai, Thailand, 2020, pp. 1-4, doi: 10.1109/iEECON48109.2020.229555.

[5] Adilakshmi, Y., & Prasad, G. V. S. N. R. V. (2019). Trust aware intrusion detection system to defend attacks in MANETs. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(6), 1298-1304.

[6] Adilakshmi Y, & G. V. S. N. R. V. Prasad. "Cooperative intrusion detection system to enhance the security in MANET." Journal of Advanced Research in Dynamical and Control Systems 11, no. 2 (2019): 100-109.

[7] Guo, Y., Zhang, H., Zhang, L., Fang, L., & Li, F. (2019). A game theoretic approach to cooperative intrusion detection. Journal of Computational Science, 30, 118-124.

[8] B. S. M. Y, R. Ibrahim, and A. Amiruddin, "Lightweight method for detecting fake authentication attack on Wi-Fi," in 2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), pp. 280-285, 2019. doi: 10.23919/EECSI48112.2019.8976975.

[9] A. S. A. Alghamdi, S. R. Hasan, M. F. Hassan, and M. S. Ali, "Jamming and anti-jamming techniques in wireless sensor networks: a comprehensive study," Wireless Personal Communications, vol. 105, no. 2, pp. 581-616, 2019.

[10] Bhunia, S., Regis, P.A., & Sengupta, S. (2018). Distributed Adaptive Beam Nulling to Survive Against Jamming in 3D UAV Mesh Networks. Computer Networks, 133, 153-166. DOI: 10.1016/j.comnet.2018.03.011

[11] Alotaibi B, & Elleithy K,"Rogue access point detection: Taxonomy, challenges, and future directions" Wireless Personal Communications, 2016, 90(3), 1261–1290.

[12] Kao K. F, Chen W. C, Chang J. C, & Chu, H. T, "An accurate fake access point detection method based on deviation of beacon time interval", 2014 IEEE Eighth international conference on software security and reliability-companion (pp. 1–2), http://dx.doi.org/10.1109/SERE- C.2014.13.

[13] Jadhav P. N, &Patil, B. M, "Low-rate DDoS attack detection using optimal objective entropy method", International Journal of Computer Applications, 2013, 78(3).

[14] N. Sufyan, N. A. Saqib, and Z. Muhammad, "Detection of jamming attacks in 802.11b wireless networks", EURASIP Journal on Wireless Communications and Networking, vol. 2013, article 208, 2013.

[15] Khairnar, V. D, & Kotecha, K,"Simulation-Based Performance Evaluation of Routing Protocols in Vehicular Ad-hoc Network" International Journal of Scientific and Research Publications, 2013, 3(10), ISSN: 2250-3153.

[16] T. Cheng, P. Li, S. Zhu, "An algorithm for jammer localization in wireless sensor networks", Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA), Fukuoka, Japan, March 2012.

[17] D. Torrieri, S. Zhu, S. Jajodia, "Moving Target Defense: Application of Game Theory and Adversary Modeling", Springer, 2012, pp. 87–96 (Chapter Cyber Maneuver Against External Adversaries and Compromised Nodes).

[18] Han C, In-Jang J, feng Shao J, Chae K, Seong-Soo B, & Jung S,"A scheme of detection and prevention rogue AP using comparison security condition of AP", 2012.

[19] Chumchu P, Saelim T, &Sriklauy C,"A new MAC address spoofing detection algorithm using PLCP header", In The International conference on information networking 2011 (pp. 48–53), IEEE.

[20] H. Liu, Z. Liu, Y. Chen, W. Xu, "Determining the position of a jammer using a virtual-force iterative approach", Wireless Networks. 17 (2) (2011) 531–547.

[21] XiangY, Li K, & Zhou W,"Low-rate DDoS attacks detection and traceback by using new information metrics", IEEE transactions on information forensics and security, 2011, 6(2), 426-437.

[22] Arackaparambil C, Bratus S, Shubina A, &Kotz D,"On the reliability of wireless fingerprinting using clock skews", In Proceedings of the third ACM Conference on wireless network security (pp. 169–174), 2010.

[23] Hamieh A, Ben-othman J, “Detection of Jamming Attacks in Wireless Ad Hoc Networks Using Error Distribution”, in IEEE International Conference on Communications, 2009 ICC '09, pp.1-6, 14-18 June 2009.

[24] J. Blumenthal, R. Grossmann, F. Golatowski, D. Timmermann, “Weighted centroid localization in zigbee-based sensor networks”, Proceedings of the IEEE International Symposium on Intelligent Signal Processing (WISP 2007), October 2007.

[25] Guo F, &Chiueh T,“Sequence number-based MAC address spoof detection. In International workshop on recent advances in intrusion detection” (pp. 309–329), 2005, Springer.

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
Seshadri Rao Knowledge Village, Gudlavalleru

Department of Computer Science and Engineering

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions., component, or software to meet the desired needs.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1 : Design, develop, test and maintain reliable software systems and intelligent systems.

PSO2 : Design and develop web sites, web apps and mobile apps.

PROJECT PROFORMA

Classification of Project	Application	Product	Research	Review
	√			

Note: Tick Appropriate category

Project Outcomes	
Course Outcome (CO1)	Identify and analyze the problem statement using prior technical knowledge in the domain of interest.
Course Outcome (CO2)	Design and develop engineering solutions to complex problems by employing systematic approach.
Course Outcome (CO3)	Examine ethical, environmental, legal and security issues during project implementation.
Course Outcome (CO4)	Prepare and present technical reports by utilizing different visualization tools and evaluation metrics.

Mapping Table

CS1537 : MAIN PROJECT															
Course Outcomes	Program Outcomes and Program Specific Outcome														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12		PSO 1	PSO 2
CO1	3	3	1					2	2	2				1	1
CO2	3	3	3	3	3			2	2	2		1		3	3
CO3	2	2	3	2	2	3	3	3	2	2	2			3	
CO4	2		1		3				3	3	2	2		2	2

Note: Map each project outcomes with POs and PSOs with either 1 or 2 or 3 based on level of mapping as follows:

1-Slightly (Low) mapped
2-Moderately (Medium) mapped
3-Substantially (High) mapped

International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169

IJRITCC International Journal
Review Report

Paper ID: IJRITCC_March_2023_5006

Dear Author/Authors

We are pleased to inform you that your manuscript titled "**ANALYSIS OF BEHAVIORAL CHARACTERISTICS OF JAMMERS TO DETECT MALICIOUS NODES IN MOBILE ADHOC NETWORKS ''**is accepted for publication in the International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC).

Your paper is peer reviewed by our expert panel and they found:

REVIEW SHEET: GENERAL JUDGMENTS

S.N.	JUDGMENT CRITERIA	COMMENT
1.	Originality of Article	90%
2.	Scope of the Article	77%
3.	Is the subject of the article within the scope of journal?	Yes
4.	Are the interpretations and conclusions sound and justified by the data?	Yes
5.	Is this a new and original contribution?	Yes
6.	Does the title of this paper clearly and sufficiently reflect its contents?	Yes
7.	Are the presentation, organization and length satisfactory?	Yes
8.	Can you suggest any reductions in the paper, or deletions of parts?	No
9.	Are the illustrations and tables necessary and acceptable?	Yes
10.	Are the references adequate and accurate, and are they all necessary?	Yes

REVIEW RESULTED: ACCEPTED

REVIEW PROCESS

Contributions submitted to IJRITCC should not have been previously published nor be currently under consideration for publication elsewhere. All articles are pre-reviewed by the editor, and if appropriate, sent for blind peer review. The editor asks the opinion of three referees who are experts in the relevant field of research. There are three referees for each submitted paper, and a minimum of two reviews related to subject is required.

Regards,
Chief-in-Editor
International Journal on Recent and Innovation Trends in Computing and Communication
https://ijritcc.org